XIV SIMPÓSIO BRASILSEIRO EM SEGURANÇA DE INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS

▶ANAIS

SBC- SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

SBSeg 2014 - Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

de 3 a 6 de Novembro de 2014 – Belo Horizonte, MG

ANAIS

Editora
Sociedade Brasileira de Computação – SBC

Organizadores
Jeroen van de Graaf, UFMG
José Marcos Nogueira, UFMG
Leonardo Barbosa Oliveira, UFMG

Realização
Universidade Federal de Minas Gerais – UFMG

Promoção
Sociedade Brasileira de Computação – SBC

Edição: Vitor Mendes Paisante

Dados Internacionais de Catalogação na Publicação

S612 Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (14. : 2014 : Belo Horizonte)
Anais / XIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais ; coordenação: Jeroen van de Graaf, José Marcos Nogueira, Leonardo Barbosa Oliveira — Porto Alegre: Sociedade Brasileira de Computação, 2014.
xviii, 753 p. il. 23 cm.

ISSN: 2176-0063

1. Ciência da computação. 2. Informática. 3. Segurança da informação. 4. Segurança de sistemas. I. Graaf, Jeroen. II. Nogueira, José. III. Oliveira, Leonardo. IV. Título.

CDU 004(063)

Mensagem da Coordenação Geral

Sejam bem-vindos ao XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2014). O SBSeg é o maior evento na área de Segurança Digital e é anualmente promovido pela Sociedade Brasileira de Computação (SBC). Este ano o simpósio ocorrerá, pela primeira vez, em Belo Horizonte -MG, sob a organização do Departamento de Ciência da Computação (DCC) da Universidade Federal do Minas Gerais (UFMG).

Será uma reunião de centenas de participantes para debater questões relativas à pesquisa, desenvolvimento e inovação no campo da Segurança Digital; entre eles, discentes, pesquisadores, docentes e profissionais do Brasil e exterior, bem como representantes da indústria, polícias e forças armadas. No entanto, os desdobramentos do SBSeg transcendem qualquer relato que possamos realizar aqui, nesta mídia. Como testemunhamos recentemente, a Segurança Digital perpassa, hoje, por questões tais como a defesa nacional e soberania de um país.

Na sua décima quarta edição, o SBSeg abrangerá 8 (oito) sessões técnicas com a apresentação de artigos que carregam consigo avanços do estado do arte em tópicos como Criptografia, Controle de Acesso e Detecção e Prevenção de Ataques. Haverá também 4 (quatro) minicursos voltados para a formação de recursos humanos de graduação e pós-graduação abordando questões como Malwares, Tolerância a Falhas e Segurança em Redes Veiculares.

O evento também abarcará uma combinação de eventos satélites jamais vista. A 8ª edição do Workshop de Trabalhos de Iniciação Científica e de Graduação (WTICG) focalizará aqueles que são o futuro da Segurança Digital no país. A 4ª edição do Workshop de Gestão de Identidades Digitais (WGID) focalizará a Gestão de Identidade Eletrônica e Identificação Civil no Brasil. A 3ª edição do Workshop de Forense Computacional (WFC) reunirá protagonistas do foro judicial brasileiro. A 1ª edição do Workshop de Tecnologia Eleitoral (WTE) abordará temas como a suposta infalibilidade da urna eletrônica brasileira e o que a academia tem a contribuir nesta frente. O 3º Concurso de Teses e Dissertações em Segurança (CTDSeg) avaliará os trabalhos de mestrado e doutorado de maior relevância científica. E, não menos importante, o Fórum de Segurança Corporativa (FSC), que objetiva aproximar atores do meio público e privado.

O SBSeg 2014 conta, ainda, com 5 (cinco) palestrantes internacionalmente reconhecidos em sua trilha principal. As palestras J. Alex Halderman, Marcus Lahr, René Peralta, Rodrigo Branco e Pascal Urien versarão sobre assuntos que vão de criptossistemas para o aumento da privacidade, passando pelo controle de acesso em Computação em Nuvem, à formas de incrementar-se a segurança de urnas eletrônicas.

Cumpre lembrar que o congresso jamais seria realizado sem a infinidade de apoio que recebeu de inúmeras partes. Assim, o nosso muito obrigado aos: participantes, autores

de trabalhos, patrocinadores, coordenadores, membros do comitê de organização, coordenadores de eventos satélites, voluntários, revisores, coordenadores e demais membros do comitê de programa, Comissão Especial de Segurança da Informação e Sistemas Computacionais (CESeg), Sociedade Brasileira de Computação, Instituto de Ciências Exatas da UFMG, Departamento de Ciência da Computação da UFMG, sua secretaria e Centro de Recursos Computacionais, alunos, professores e funcionários da UFMG, prestadores de serviço, recursos humanos do Centro de Atividades Didáticas de Ciências Humanas da UFMG e todos os demais que possibilitaram a realização do evento.

Enfim, o Comitê Organizador do SBSeg 2014 deseja uma semana bastante profícua, fecunda e, por que não, divertida. Ademais, desejamos que o SBSeg semeie inúmeros avanços científicos e tecnológicos para que de forma mais célere transformemos o nosso país, o Brasil, em uma nobre nação.

Jeroen van de Graaf, UFMG
José Marcos Nogueira, UFMG
Leonardo Barbosa Oliveira, UFMG
Coordenadores Gerais do SBSEG 2014

Mensagem da Coordenação do Comitê de Programa

O Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg) é o maior evento acadêmico brasileiro na área de Segurança Computacional. É sempre desafiador construir um programa que atenda aos critérios de excelência acadêmica e aos mais diversos interesses dos participantes do evento, especialmente quando considerada a sua história e reputação.

Para a décima quarta edição do simpósio, seguimos a tradição estabelecida e dividimos o programa em artigos completos e resumos estendidos. Os artigos completos apresentam avanços no estado-da-arte em segurança, apoiados por resultados concretos, enquanto os resumos estendidos se baseiam em resultados preliminares. Neste ano, optamos por dois processos seletivos independentes, cada qual com calendário próprio: primeiro foram aceitas submissões e selecionados artigos completos, e posteriormente, de resumos estendidos. Em resposta à chamada de trabalhos, foram recebidos para avaliação (em meio à Copa do Mundo no Brasil) 92 submissões, sendo 66 artigos completos e 26 resumos. Destes, foram aceitos para publicação e apresentação no evento 22 artigos e 11 resumos estendidos.

Buscando melhorar ainda mais a qualidade técnica do evento, fizemos com que o processo de revisão e seleção contasse com 5 pareceres por submissão de artigo completo e 3 revisões por resumo estendido, e que cada artigo em conflito fosse discutido. Nesse aspecto, a colaboração pronta e competente do corpo de revisores foi essencial, garantindo que ao final cada artigo e resumo tivessem respectivamente pelo menos 4 e 2 revisões cada. O programa resultante possui seis sessões técnicas, nos temas de Controle de Acesso, Ciência Forense, Criptografia, Segurança em Redes, Prevenção de Ataques e Segurança de Aplicações; e duas sessões técnicas dedicadas à apresentação dos resumos estendidos.

Agradecemos à Coordenação Geral do SBSeg 2014, bem como aos membros da Comissão Especial de Segurança da Informação e de Sistemas Computacionais (CE-Seg) da Sociedade Brasileira de Computação (SBC), por nos confiar a Coordenação do Comitê de Programa do evento. Somos gratos também ao Comitê de Programa, pela sua dedicação e qualidade das revisões, mesmo frente a uma carga superior comparada a edições anteriores. Em particular, agradecemos aos autores que submeteram artigos completos e resumos estendidos, sem os quais o evento não seria um sucesso.

Por fim, esperamos ter contribuído para a formação de um programa de excelência e desejamos a todos os participantes um evento interessante e produtivo, repleto de oportunidades para expandir horizontes e forjar parcerias e colaborações de pesquisa.

Diego F. Aranha, UNICAMP
Marinho P. Barcellos, UFRGS
Coordenadores do Comitê de Programa do SBSeg 2014

Mensagem da Coordenação do WTICG

Sejam bem vindos ao VIII Workshop de Trabalhos de Iniciação Científica e de Graduação (WTICG). O WTICG é um evento que ocorre anualmente, sendo promovido pela Sociedade Brasileira de Computação (SBC). A sua oitava edição ocorre neste ano de 2014 em Belo Horizonte-MG, em conjunto com o Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG).

O WTICG foi criado para dar oportunidade a alunos de graduação ou recém-graduados para apresentar seus trabalhos científicos sobre temas relacionados às áreas de segurança da informação e de sistemas computacionais.

Nesta oitava edição do WTICG foram aceitos 12 artigos científicos de boa qualidade técnica envolvendo temas tanto de segurança de sistemas computacionais como de segurança da informação. Tal como nos anos anteriores, uma comissão irá avaliar as apresentações durante o workshop e selecionar os três melhores artigos do WTICG'14. Gostaria de agradecer aos membros do comitê de programa e avaliadores convidados pela competência e dedicação na avaliação desses artigos.

Um agradecimento especial também vai aos coordenadores gerais deste XIV SBSEG – Jeroen van de Graaf, José Marcos Nogueira e Leonardo B. Oliveira – pela confiança depositada e pelo apoio prestado ao longo de todo o processo. Além disso, estendo o nosso agradecimento à equipe de suporte que mantém o sistema eletrônico JEMS. Por fim, agradeço aos autores de artigos e participantes do Workshop pelo seu interesse no evento, bem como pelo trabalho investido no aprimoramento dos artigos e na preparação das apresentações.

Eduardo Souto
Coordenador do Comitê de Programa WTICG'14

Mensagem da Coordenação do WGID

O Workshop de Gestão de Identidades Digitais (WGID) é um evento integrante do SBSeg que objetiva ser um fórum para discussões em torno do estado da arte de tecnologias relacionadas à gestão de identidades digitais. Além disso, ele busca identificar os desafios de pesquisa e aproximar os grupos atuantes na área.

Nesta edição, o programa do Workshop iniciará com um painel de discussões sobre o estágio atual e as próximas etapas do projeto do Ministério da Justiça para criação de uma nova cédula de identidade única para todo o país, projeto conhecido como Registro de Identificação Civil ou RIC. Além das tradicionais informações impressas na cédula, haverá também um chip que permitirá ativar funções avançadas de autenticação, como o uso de certificação digital, por exemplo.

Em seguida teremos duas breves apresentações sobre as atividades do Comitê Técnico de Gestão de Identidades e sobre o Laboratório de Gestão de Identidades, ambos da RNP, com o objetivo de divulgar os serviços oferecidos pelos mesmos para a comunidade de ensino e pesquisa nacionais.

Pesquisadores foram convidados a submeter trabalhos originais relacionados à Gestão de Identidades, o que resultou na submissão de vários textos e na seleção de alguns artigos completos e outros curtos que serão apresentados na sessão técnica.

Gostaríamos de agradecer aos membros do Comitê de Programa do WGID pela qualidade do trabalho realizado nas revisões dos artigos. Registramos um agradecimento especial a todos os autores que prestigiaram o WGID ao submeterem trabalhos relatando suas pesquisas.

Gostaríamos também de agradecer a todos que colaboraram na organização do WGID 2014, em especial, aos Coordenadores Gerais do SBSeg 2014, Profs. Jeroen van de Graaf, José Marcos Nogueira e Leonardo Barbosa Oliveira.

Em nome do WGID, saudamos todos os participantes deste workshop, com votos de que aproveitem ao máximo suas atividades, bem como os demais eventos do SBSeg 2014.

Prof. Marco Aurélio Amaral Henriques
Coordenador do WGID 2014

SBSeg – III Workshop de Forense Computacional (WFC) Mensagem dos Coordenadores

O Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg) abriu as portas para a área interdisciplinar de Forense Computacional e o **Workshop de Forense Computacional (WFC)** já está em sua 3ª edição! Os eventos de 2012 (Curitiba-PR) e 2013 (Manaus-AM) foram realizados com sucesso, visto o nível dos palestrantes, sessões técnicas e artigos selecionados. Durante o ano de 2014, trabalhamos para que esta 3ª edição do WFC também seja um sucesso.

Esperamos que o III WFC, em Belo Horizonte, contribua para formação e motivação da comunidade, visto existirem muitas deficiências relacionadas à área de Forense Computacional, tais como: formação de recursos humanos especializado, padronização de métodos e técnicas, desenvolvimento de ferramentas (hardware e software), ausência de fóruns de discussão e trocas de experiências e, ainda, recursos financeiros para Pesquisa, Desenvolvimento e Inovação.

O III WFC sozinho não irá resolver todos os problemas, mas fomentará a discussão dos problemas e das possíveis soluções técnico-científicas que podem auxiliar os peritos e profissionais na atividade cotidiana. O evento representa a troca de conhecimentos nas áreas de Ciência da Computação e Direito, trazendo técnicas, métodos e segurança necessários em todos os tipos de trabalhos, operações, perícias, investigações e atividades relacionadas com a Forense Computacional. Sem esquecer, dos aspectos jurídicos, legislativos e implicações à sociedade brasileira decorrentes de atos ilícitos.

O **WFC 2014** reúne renomados profissionais para ministrar palestras incluindo: *International Federation on Intellectual Property* (IFIP), Polícia Científica do Paraná – Instituto de Criminalísta do Paraná, Micro Systemation, Imagina Lab da UFRN e 7º. Tabelionato Volpi – Curitiba. Os temas tratados abordam: *Aspectos Juridicos da Forense Computacional*, *Computação Forense Permeando todas as Áreas da Perícia, Mini Computadores, Implications of ageing in biometrics system design e A imputação de Autoria em Documentos Digitais*.

Além disto, marcam presença no evento profissionais atuantes na área de Direito Digital, Tecnologia da Informação e Segurança. Foram ainda selecionados seis artigos para apresentação oral durante o evento, formando uma grade motivadora de apresentações.

Agradece-se em especial ao Comitê de Programa que auxiliou na avaliação dos artigos submetidos e, conclui-se, portanto, que a Forense Computacional desempenha papel importante e vem se firmando como uma peça fundamental do quebra-cabeça da investigação. Saudamos todos os palestrantes e partipantes do WFC com os votos de um excelente Workshop e uma ótima estadia em Belo Horizonte!

Profa. Dra. Cinthia O. A. Freitas, PUCPR, PR
Prof. Dr. William Robson Schwartz, UFMG, MG
Coordenadores do WFC

Comitê de Programa do WFC

Cinthia O. A. Freitas, PUCPR, PR
William Robson Schwartz,
DCC/UFMG, MG
Alceu de Souza Britto Jr, PUCPR, PR
Luiz Eduardo Soares de Oliveira, UFPR, PR

Mensagem da Coordenação do WTE

A primeira edição do Workshop de Tecnologia Eleitoral (WTE), como evento satélite do XIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg 2014), tem como objetivo principal fomentar a pesquisa em tecnologia eleitoral, na tentativa de reaproximar a academia das questões ligadas ao voto eletrônico. Apesar do Brasil realizar eleições puramente eletrônicas desde 2000, a participação do meio acadêmico, em particular do segmento orientados a Segurança Computacional, tem sido praticamente inexistente na iniciativa. O problema ganha contornos críticos após sucessivas manifestações da comunidade científica brasileira na direção de apontar problemas intrínsecos no modelo de votação eletrônica utilizado no Brasil. Desta forma, o treinamento de pesquisadores em tecnologia eleitoral é fundamental para a análise crítica e incremento de segurança do sistema de votação eletrônica em uso no país e também de outros sistemas propostos na literatura científica. A realização de um *workshop* com frequência regular facilita não apenas a formação de novos pesquisadores, mas o contato mais próximo de estudantes e profissionais com o estado da arte em tecnologia eleitoral. Além disso, reúne os principais pesquisadores brasileiros na área em um fórum natural de discussão técnica.

Para a edição inicial, decidiu-se por um evento curto e de organização simples, com programa formado pela apresentação de artigos científicos, apresentações de trabalhos em curso por membros do Comitê de Programa e uma palestra convidada ministrada por pesquisador reconhecido internacionalmente na área. Foram recebidas e revisadas 5 submissões em caráter anônimo, sendo aceitas para publicação as 4 submissões melhor avaliadas. A palestra internacional, ministrada pelo professor J. Alex Halderman da Universidade de Michigan, trata da análise de segurança de sistemas de votação eletrônica via Internet. Observado o interesse geral no tema e sua relação com outros aspectos em Segurança da Informação, a palestra também compõe a programação do SBSeg, contribuindo para difusão de conhecimento na área de tecnologia eleitoral.

A coordenação do evento agradece em especial ao Comitê de Programa, formado pelos professores Mario Gazziro (UFABC), Ruy J. Guerra B. de Queiroz (UFPE) e Roberto Samarone Araújo (UFPA); que auxiliou na avaliação dos artigos submetidos e contribuiu com a composição do programa com palestras sobre seus trabalhos em curso. Espera-se que o *workshop* venha desempenhar papel importante do ponto de vista científico, na forma de apresentação de trabalhos acadêmicos; mas também contribua para um amadurecimento na discussão em torno de tecnologia eleitoral utilizada no país, estabelecendo bases científicas para sua avaliação. Que os participantes do WTE tenham todos um evento interessante e produtivo!

Coordenadores do WTE
Prof. Dr. Diego F. Aranha, UNICAMP
Prof. Dr. Jeroen van de Graaf, UFMG

Mensagem da Coordenação do CTDSeg

O Concurso de Teses e Dissertações em Segurança da Informação e Sistemas Computacionais (CTDSeg), evento integrante do SBSeg, visa selecionar e premiar as melhores teses de doutorado e as melhores dissertações de mestrado na área concluídas nos últimos dois anos.

Os pesquisadores foram convidados a submeter um artigo descrevendo e resumindo os resultados obtidos em suas teses e dissertações concluídas no período de agosto de 2012 a junho de 2014. Nesta edição, dezoito artigos descrevendo os resultados de dissertações de mestrado ou de teses de doutorado foram submetidos ao CTDSeg 2014.

Os artigos submetidos passaram por uma etapa inicial de avaliação que selecionou as cinco melhores teses de doutorado e as cinco melhores dissertações de mestrado. Cada artigo foi avaliado pelos membros do comitê de avaliação considerando os méritos científicos. Os trabalhos selecionados serão apresentados em sessões no dia 03 de novembro de 2014, durante a programação do SBSeg 2014, em Belo Horizonte MG.

A tarefa da coordenação do processo de seleção de trabalhos científicos foi uma atividade muito gratificante para nós. Desta tarefa, participaram 12 revisores que, de forma voluntária, avaliaram os artigos submetidos. Gostaríamos de agradecer o empenho dos membros do comitê de avaliação pela alta qualidade do trabalho realizado. Registramos um agradecimento especial a todos os autores que prestigiaram o CTDSeg 2014 ao submeterem trabalhos relatando suas pesquisas. Sem os autores e seu interesse pelo evento não teríamos um evento com trabalhos de tanta qualidade.

Gostaríamos também de agradecer a todos que colaboraram na organização do CTDSeg 2014, especialmente, à Comissão Especial em Segurança da Informação e de Sistemas Computacionais da SBC que vem incentivando continuamente a realização do evento; aos coordenadores do SBSeg 2014, Profs. Jeroen van de Graaf, José Marcos Nogueira e Leonardo Barbosa Oliveira, e toda a equipe do comitê local pelo apoio técnico, logístico e de divulgação dedicado à realização desta edição do CTDSeg. Em nome do Comitê de Avaliação, saúdo a todos os participantes do CTDSeg 2014, com votos de um evento bastante profícuo.

Aldri Luiz dos Santos, UFPR
Anderson C. A. Nascimento, UnB
Coordenadores do Concurso de Teses e Dissertações do SBSeg 2014

Comitês

Comitê de Organização

Coordenação Geral

- Jeroen van de Graaf (UFMG)
- José Marcos Nogueira (UFMG)
- Leonardo B. Oliveira (UFMG)

Coordenação do Comitê de Programa

- Diego F. Aranha (UNICAMP)
- Marinho P. Barcellos (UFRGS)

Coordenadores do CTD

- Aldri Santos (UFPR)
- Anderson do Nascimento (UNB)

Coordenador do FSC

- Pericles Luz

Coordenador de Minicursos

- André dos Santos (UECE)

Coordenador de Palestras e Tutoriais

- Michele Nogueira (UFPR)

Coordenadores do WFC

- Cinthia Freitas (PUCPR)
- William Robson Schwartz (UFMG)

Coordenador do WGID

- Marco Aurélio Amaral Henriques (UNICAMP)

Coordenadores do WTE

- Diego F. Aranha (UNB)
- Jeroen van de Graaf (UFMG)

Coordenador do WTICG

- Eduardo Souto (UFAM)

Publicidade

- Mário S. Alvim (UFMG)

Comitê de Programa

- Adriano Cansian (UNESP)
- Alberto Egon Schaeffer Filho (UFRGS)
- Aldri dos Santos (UFPR)
- Altair Santin (PUC-PR)
- Anderson Nascimento (UnB)
- André dos Santos (UECE)
- André Gregio (CTI)
- Antonio Augusto de Aragão Rocha (UFF)
- Arlindo L. Marcon Jr. (IFPR)
- Carla Merkle Westphall (UFSC)
- Carlos Maziero (UTFPR)
- Carlos Westphall (UFSC)
- Célio Albuquerque (UFF)
- Diego de Freitas Aranha (UNICAMP)
- Djamel Fawzi Hadj Sadok (UFPE)
- Eduardo Feitosa (UFAM)
- Eduardo Souto (UFAM)
- Emerson Ribeiro de Mello (IFSC)
- Eulanda Miranda dos Santos (UFAM)
- Fernando Magno Quintao (UFMG)
- Hao Chi Wong (Intel)
- Jean Martina (UFSC)
- Jeroen van de Graaf (UFMG)
- Joaquim Celestino Júnior (UECE)
- Joni da Silva Fraga (UFSC)
- Julio López (UNICAMP)
- Lau Cheuk Lung (UFSC)
- Leonardo B. Oliveira (UFMG)
- Lisandro Zambenedetti Granville (UFRGS)
- Luciano Paschoal Gaspary (UFRGS)
- Luiz Carlos Albini (UFPR)
- Luiz Fernando Rust da Costa Carmo (UFRJ)
- Marcos Simplicio (Poli-USP)
- Marinho P. Barcellos (UFRGS)
- Mário Alvim (UFMG)
- Michele Nogueira (UFPR)
- Michelle Wangham (UNIVALI)
- Paulo André da Silva Gonçalves (UFPE)
- Paulo Lício de Geus (UNICAMP)
- Paulo S. L. M. Barreto (Poli-USP)
- Pedro Velloso (UFF)
- Raul Ceretta Nunes (UFSM)
- Raul Weber (UFRGS)
- Ricardo Custódio (UFSC)

- Ricardo Dahab (UNICAMP)
- Roberto Gallo (KRYPTUS)
- Routo Terada (USP)
- Ruy José Guerra Barretto de Queiroz (UFPE)
- Sergio de Oliveira (UFSJ)

CESeg

Coordenadores

- Anderson C. A. Nascimento (UnB, coordenador)
- Aldri L. Dos Santos (UFPR, vice)

Comitê consultivo

- Anderson Clayton Alves Nascimento (UnB)
- Aldri Luiz dos Santos (UFPR)
- Jeroen van de Graaf (UFMG)
- Ricardo Dahab (UNICAMP)
- Altair Santin (PUC-PR)
- Eduardo Souto (UFAM)
- Raul Ceretta Nunes (UFSM)
- Michele Nogueira (UFPR)
- Diego Aranha (UNICAMP)

Colaboradores

- Artur Luis (UFMG)
- Júlia Terra (UFMG)
- Vitor Paisante (UFMG)

Sumário

Trilha Principal (artigos completos)

I. Controle de acesso

II. Forense

II. Criptografia

Trilha Principal (resumos estendidos)

WTICG - Workshop de Trabalhos de Iniciação Científica e de Graduação

WGID - Workshop de Gestão de Identidades

CTDSeg - Concurso de Teses e Dissertações em Segurança

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

Trilha principal — Artigos Completos

Controle de acesso baseado em reencriptação por *proxy* em Redes Centradas em Informação

Elisa Mannes[1], Carlos Maziero[1], Luiz Carlos Lassance[2], Fábio Borges[3]

[1]Programa de Pós-graduação em Informática – Universidade Federal do Paraná (UFPR)

[2]Confederação das Cooperativas Centrais de Crédito Rural com Interação Solidária (CONFESOL)

[3]Technische Universität Darmstadt/CASED, Telecooperation Group

elisam@inf.ufpr.br, maziero@utfpr.edu.br

luiz@confesol.com.br, fabio.borges@cased.de

Abstract. *Information-centric networks (ICN) represent a promising approach to the Future Internet, addressing the shortcomings of the current Internet with a suitable infrastructure for content distribution. By naming, routing, and forwarding content instead of machine addresses, the ICN shift the protagonists at the network layer from hosts to contents. One implication is the in-network cache, which allows a better use of communication channels and faster delivery of content to the user. However, the ability to receive content from caches generates concerns about access control. In this context, we propose a solution for access control in ICN based on proxy re-encryption. The proposed solution ensures that only authorized users are able to access content, while maintaining the beneficial effects of caching in ICN, even in face of malicious entities.*

Resumo. *As redes centradas em informação (ICN) representam uma abordagem promissora para a Internet do Futuro, pois aborda as atuais deficiências da Internet com uma infraestrutura mais adequada para a distribuição de conteúdo. Ao nomear, rotear e encaminhar conteúdo ao invés de endereços de máquina, a ICN desloca o protagonismo da camada de rede das máquinas para os conteúdos. Uma das implicações dessa mudança é o cache nos dispositivos de rede, que permite uma melhor utilização dos canais de comunicação e uma entrega mais rápida do conteúdo ao usuário. Entretanto, a possibilidade de receber conteúdo dos caches gera preocupações com relação ao controle de acesso. Neste contexto, propõe-se uma solução para controle de acesso em ICN baseada em reencriptação por proxy. A solução proposta garante que somente usuários autorizados acessem o conteúdo na rede enquanto se mantém os benefícios do sistema de cache em ICN, mesmo diante de uma entidade maliciosa.*

1. Introdução

As redes centradas em informação (ICN - *Information-centric Networks*) [Ahlgren et al. 2012, Brito et al. 2012] propõem superar as dificuldades atuais da Internet modificando a principal entidade da rede de máquinas para conteúdos. Esse novo paradigma traz características especiais para a Internet, pois nomear, rotear e encaminhar conteúdo na rede ao invés de endereços de máquina permite a implementação de *cache*

nos dispositivos da rede, por exemplo. Um mecanismo de *cache* diretamente na rede potencializa um melhor desempenho na entrega do conteúdo e torna a arquitetura mais adequada para os atuais padrões de tráfego, inclusive para dispositivos móveis. Entretanto, o paradigma de ICN também modifica os aspectos relacionados à segurança de redes. Por exemplo, a nomeação dos conteúdos exige que mecanismos de segurança sejam focados no conteúdo ao invés de prover segurança para máquinas, *links* e sessões. Além do mais, o emprego de *caches* na rede resulta em conteúdos recuperados de qualquer dispositivo por qualquer usuário, trazendo novos desafios com relação à **privacidade** e ao **controle de acesso**, já que não é obrigatório que o usuário se conecte ao provedor de conteúdo para requisitar o conteúdo. Esse problema toma proporções ainda maiores quando considerados os ambientes móveis, em que qualquer dispositivo pode rotear e armazenar conteúdos em *cache*, incluindo dispositivos maliciosos ou comprometidos.

As soluções atuais para controle de acesso na distribuição de conteúdo, apesar de serem transferíveis para ICN, geralmente inviabilizam a proposta do uso de *cache* na rede. O uso de criptografia assimétrica, por exemplo, inibe o compartilhamento das cópias em *cache* por diversos usuários, pois o conteúdo é encriptado para cada usuário individualmente. Além do mais, a arquitetura atual exige que os usuários se autentiquem em servidores específicos para garantir a segurança ao requisitar conteúdos. Novamente, essa solução prejudica a implementação de mecanismos de *cache*. Além dessas soluções tradicionais, existem soluções de controle de acesso desenvolvidas especialmente para uso em arquiteturas de ICN [Ion et al. 2013, Misra et al. 2013, Papanis et al. 2013, Fotiou et al. 2012, Singh et al. 2012, Hamdane et al. 2013]. Contudo, a maioria emprega o uso de criptografia simétrica e foca na garantia de que somente usuários autorizados tenham acesso à chave utilizada. Essa estratégia pode representar um problema caso a chave seja divulgada por uma entidade maliciosa, já que é a mesma para todos os usuários.

Neste artigo, propomos uma solução de controle de acesso para conteúdo protegido em ICN focando em três aspectos principais: (i) o conteúdo pode ser armazenado em qualquer dispositivo e recuperado por qualquer usuário; (ii) os usuários que acessam o conteúdo não podem decifrá-lo, a menos que sejam autorizados pelo provedor de conteúdo; (iii) não há a adição de novas entidades na rede para a aplicação ou a validação de políticas de acesso. A reencriptação por *proxy* [Ateniese et al. 2006] é um esquema de criptografia em que uma mensagem encriptada com uma chave pública A pode ser transformada em uma mensagem encriptada com uma chave pública B, sem expor o conteúdo original nem as chaves privadas correspondentes. Essa transformação é tradicionalmente feita por uma entidade semi-confiável denominada *proxy de reencriptação*, usando uma *chave de reencriptação* definida a partir das chaves A e B. A reencriptação por *proxy* pode potencialmente ser usada como mecanismo de controle de acesso a conteúdos em ICN da seguinte forma: o conteúdo original é encriptado com a chave pública do provedor, gerando um conteúdo encriptado único. Um usuário interessado no conteúdo pode recuperá-lo no *cache* mais próximo; em seguida ele deve interagir com o provedor, visando obter a chave de reencriptação necessária para reencriptar aquele conteúdo com sua chave pública, permitindo seu acesso. Ainda que o controle de acesso seja desejável para a maioria dos conteúdos na Internet, neste trabalho focamos as especificidades de conteúdos populares, em que um grande conjunto de usuários esteja interessado no mesmo conteúdo, tais como vídeos, *e-books*, *streaming* e atualizações de *software*. Nesses cenários, o mecanismo de *cache* é utilizado em seu potencial máximo e os benefícios da ICN surgem de uma forma

mais substancial. Entretanto, a solução proposta é aplicável para outros tipos de conteúdo.

Este artigo está organizado como segue: a Seção 2 descreve os esforços atuais para o controle de acesso em ICN e discute suas principais deficiências. A Seção 3 explica os fundamentos de reencriptação por *proxy*, que fundamenta a solução proposta. A Seção 4 detalha a proposta e descreve seus aspectos técnicos. A Seção 5 valida a solução proposta e analisa o desempenho com relação ao tempo computacional para diferentes tamanhos de mensagens e chaves. A Seção 6 discute o uso da solução proposta com relação ao desempenho, segurança e adequabilidade para a arquitetura de ICN. Por fim, a Seção 7 conclui o artigo e sugere trabalhos futuros.

2. Controle de acesso em ICN

A possibilidade de armazenamento de conteúdo nos dispositivos da rede gera uma grande preocupação com relação ao controle de acesso dos conteúdos, pois as cópias em *cache* podem ser acessadas por qualquer usuário, inclusive aqueles que não têm autorização de acesso ao conteúdo. Para serviços que requerem o pagamento de mensalidades, tais como *Netflix*, *Hulu*, *Amazon*, *Apple* e *Play Store* e *Steam*, fica ainda mais evidente a necessidade de assegurar o controle de acesso para conteúdo armazenado em *cache*. Tais serviços geralmente requerem um rigoroso controle das contas de usuários, do número de reproduções do conteúdo e da quantidade de dispositivos autorizados, por exemplo. Restringir o conhecimento do nome do conteúdo somente para os usuários autorizados não é suficiente, já que em ICN as ações de roteamento e de encaminhamento são realizadas diretamente pelo nome do conteúdo e, desta forma, os nomes dos conteúdos podem ser facilmente descobertos. Assim, esse tipo de aplicação requer uma solução de controle de acesso mais robusta e adequada para o uso em ICN. De outra forma, é pouco provável que a arquitetura de ICN seja adotada para a distribuição de conteúdos protegidos.

A encriptação do conteúdo é apontada como a ação mais básica para garantia que somente usuários autorizados, que possuam uma chave válida, possam acessá-lo [Jacobson et al. 2012]. Contudo, a encriptação de um mesmo conteúdo para usuários diferentes não é conveniente, pois o conteúdo encriptado para um usuário específico não pode ser acessado por outro e, portanto, as cópias armazenadas em *cache* não são aproveitadas. Como alternativa, outras abordagens foram propostas. Uma delas propõe a encriptação do conteúdo com uma chave simétrica, aproveitando o mecanismo de *cache*, enquanto as chaves simétricas e as licenças são encriptadas para grupos de usuários. Exemplos de tais abordagens são as soluções propostas por [Misra et al. 2013] e [Papanis et al. 2013], que exploram a criptografia de *broadcast* e a baseada em atributos, respectivamente. Contudo, essas soluções protegem parcialmente o conteúdo, pois o uso da criptografia simétrica pode representar um ponto de vulnerabilidade no caso de a chave simétrica ser divulgada na rede. Uma exceção é a solução proposta por [Ion et al. 2013], em que o conteúdo é encriptado de acordo com atributos e as políticas de acesso são aplicadas pelo próprio texto encriptado ou pela chave do usuário. Entretanto, essa abordagem encripta o conteúdo por grupos de atributos e, desta forma, o conteúdo em *cache* pode não servir a todos os usuários. Outras soluções propostas [Fotiou et al. 2012, Singh et al. 2012, Hamdane et al. 2013] requerem o uso de servidores terceirizados para aplicar as políticas de acesso, que além de necessitar de uma infraestrutura terceira, pode depender de peculiaridades das arquiteturas de ICN.

A reencriptação por *proxy* já foi explorada no contexto de controle de

acesso a conteúdos por [Xiong et al. 2012, Kissel and Wang 2013, Wood and Uzun 2014]. [Xiong et al. 2012] apresenta uma solução em que o provedor de conteúdo cria grupos com um respectivo par de chaves pública-privada. A reencriptação ocorre através de um *proxy* localizado na nuvem. O conteúdo é dividido em duas partes e somente a menor parte é reencriptada. [Kissel and Wang 2013] também propõe que o proprietário do conteúdo crie um grupo de usuários com um respectivo par de chave pública-privada. Ao entrar no grupo, é dado ao usuário uma chave de re-encriptação que permite que ele re-encripte e posteriormente decripte os conteúdos do grupo. Para revogar o acesso de um usuário, é proposto que o grupo seja desfeito e recriado com um novo par de chaves e que seja dado aos usuários autorizados a nova chave de re-encriptação, sendo o primeiro trabalho a propor a transferência da função de *proxy* para o usuário. Esses dois trabalhos consideram que os provedores de conteúdo têm controle sobre o conteúdo, podendo revogar o acesso a qualquer tempo, o que não é garantido em ICN. Paralelamente ao nosso trabalho, [Wood and Uzun 2014] explora a re-encriptação por *proxy* no contexto de ICN. Contudo, a análise conduzida pelos autores os levam a propor o uso da criptografia simétrica para encriptar os conteúdos e a aplicação da re-encriptação para a proteção de chaves simétricas, o que incorre nas mesmas deficiências apontadas anteriormente. Desta forma, observa-se que fornecer uma solução de controle de acesso para ICN não é uma tarefa trivial e requer o alinhamento de vários objetivos, principalmente com relação à segurança dos dados e ao desempenho na entrega de conteúdo através dos *caches*.

3. Reencriptação por *proxy*

A ideia geral dos esquemas tradicionais de reencriptação por *proxy* (PRE - *proxy re-encryption* [Ateniese et al. 2006]) é permitir a transformação de uma mensagem encriptada com a chave pública de um usuário A, para uma mensagem encriptada com a chave pública de um usuário B. Essa transformação acontece em uma terceira entidade considerada semi-confiável, o *proxy*. O usuário A autoriza o *proxy* a transformar as mensagens encriptadas com a sua chave pública para a chave pública do usuário B ao concedê-lo uma **chave de reencriptação** $rk_{A \to B}$. A Figura 1 ilustra o funcionamento de um esquema básico de reencriptação por *proxy* composto pelos usuários A e B e por um *proxy*. Neste exemplo, o usuário A encripta um conteúdo C com a sua chave pública $kp(A)$, gerando $\{C\}_{kp(A)}$. Caso o usuário A queira permitir que o usuário B acesse o conteúdo C, ele envia ao *proxy* o conteúdo $\{C\}_{kp(A)}$ e uma chave de reencriptação $rk_{A \to B}$, calculada com base na chave pública $kp(B)$ do usuário B. O *proxy* então utiliza a chave de reencriptação enviada por A para reencriptar o conteúdo para B, gerando $\{C\}_{kp(B)}$. O *proxy* envia o conteúdo $\{C\}_{kp(B)}$ para o usuário B, que o decripta utilizando a sua chave privada $kv(B)$.

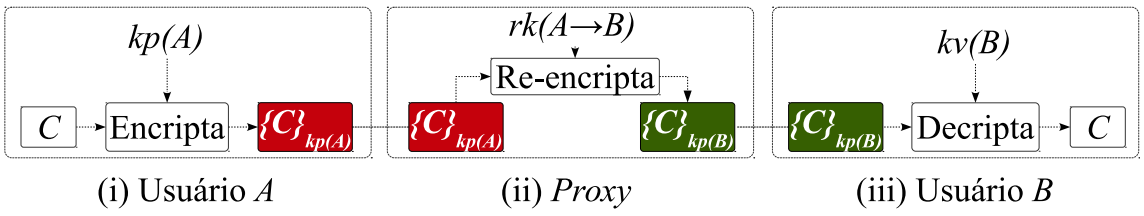


Figura 1. Visão geral do esquema de reencriptação por *proxy*

Em geral, os esquemas de reencriptação por *proxy* garantem duas asserções básicas: o *proxy* não pode ser capaz de acessar o conteúdo da mensagem que reencripta e, de

posse da mensagem encriptada e da chave de reencriptação, não pode recuperar as chaves privadas de A ou B. [Ateniese et al. 2006, Chow et al. 2010] definem os algoritmos que compõem um esquema de reencriptação por *proxy*, definidos abaixo:

CONFIGURAÇÃO: recebe como entrada um parâmetro de segurança k e tem como saída uma tupla de parâmetros globais PARAM.

GERAÇÃO DE CHAVES: gera pares de chaves pública-privada (kp, kv).

ENCRIPTAÇÃO: ao receber $kp(A)$ e uma mensagem m, gera uma mensagem encriptada $\{m\}_{kp(A)}$.

DECRIPTAÇÃO: ao receber $kv(A)$ e $\{m\}_{kp(A)}$, gera como saída a mensagem m.

GERAÇÃO DE CHAVE DE REENCRIPTAÇÃO: tem como entrada a chave privada $kv(A)$ e a chave pública $kp(B)$ e como saída uma chave de reencriptação $rk_{A\to B}$.

REENCRIPTAÇÃO: ao entrar a chave de reencriptação $rk_{A\to B}$ e o texto encriptado $\{m\}_{kp(A)}$, tem como saída $\{m\}_{kp(B)}$.

Para fundamentar a solução proposta neste artigo, são necessárias três propriedades fundamentais dos esquemas de reencriptação por *proxy* (contudo outras propriedades podem ser incorporadas para agregar outras funcionalidades):

- *Unidirecionalidade*: a delegação de direitos de decriptar de $A \to B$ não implica na delegação de $B \to A$;
- *Salto único*: somente mensagens originais podem ser reencriptadas;
- *Segurança contra conluio*: o usuário B e o *proxy* em conluio não conseguem recuperar a chave privada de A.

Com base nos esquemas de reencriptação por *proxy* tradicionais, propomos uma solução para controle de acesso em ICN. Para alinhar o esquema de reencriptação por *proxy* com as peculiaridades da ICN, a entidade *proxy* é eliminada do processo de reencriptação. Contudo, as funções tradicionalmente desempenhadas pelo *proxy* são transferidas para o usuário. Sendo assim, na solução proposta, há somente duas entidades envolvidas na reencriptação: a fonte e o usuário. Neste artigo, emprega-se o esquema de reencriptação por *proxy* proposto por [Chow et al. 2010].

4. Controle de acesso usando reencriptação

Diferentemente das abordagens existentes, nosso objetivo é propor uma solução de controle de acesso que seja alinhada ao funcionamento das arquiteturas de ICN, garantindo a disponibilidade do conteúdo em qualquer *cache* na rede enquanto permite o controle de acesso ao conteúdo pelo provedor do mesmo. Além do mais, é desejável que a solução não utilize entidades extras para o controle de acesso nem modifique as funções do núcleo de qualquer arquitetura de ICN, mantendo o processo simples e seguindo as especificações da arquitetura de ICN. Na nossa visão, o paradigma de ICN deve ser mantido simples para entregar o conteúdo da melhor forma, sem sobrecarregar os roteadores com a verificação de políticas de acesso ou criar uma nova infraestrutura de servidores para validar o acesso de usuários e serviços ao conteúdo em *cache*. Para isso, propomos a utilização de um esquema de reencriptação por *proxy* para o controle de acesso aos conteúdos em ICN. Neste esquema, os provedores de conteúdo encriptam os conteúdos com chaves públicas

correspondentes e distribuem o conteúdo na rede conforme as requisições dos usuários. Os usuários podem recuperar os conteúdos tanto do provedor de conteúdo como dos *caches*. Para decifrá-los, os usuários devem solicitar uma chave de reencriptação para o provedor de conteúdo. As próximas subseções detalham a solução proposta.

4.1. Modelo de rede

Neste trabalho, consideramos as especificidades da arquitetura NDN (*Named-Data Network* [Jacobson et al. 2012]), mas por não modificar entidades na rede, a solução proposta pode ser aplicada a qualquer arquitetura de ICN. A infraestrutura da NDN é composta por provedores de conteúdo (P), roteadores (R) e usuários (U). Os provedores de conteúdo anunciam os nomes dos seus conteúdos na rede. Cada conteúdo é dividido em *chunks* de 4Kb, que são individualmente nomeados e a ligação entre o conteúdo e o seu nome é assinada criptograficamente, para que os usuários possam validar a integridade e a autenticidade do conteúdo, conforme [Smetters and Jacobson 2009]. Para requisitar um conteúdo na rede, o usuário deve enviar um pacote de *Interesse* e, em resposta, recebe um pacote de *Dados* com o *chunk* solicitado. Os nomes de conteúdos vindos do mesmo provedor de conteúdo compartilham prefixos em comum que permitem a agregação nas tabelas de roteamento nos roteadores.

Além de rotear e encaminhar os conteúdos nomeados para os usuários, os roteadores têm a função de armazenar conteúdo em seus *caches* de acordo com as políticas de *cache*, desta forma permitindo um melhor desempenho na entrega dos conteúdos. Ao receber um pacote de *Interesse*, o roteador verifica seu *cache* (*CS - content store*). Caso o conteúdo solicitado esteja armazenado no *cache*, ele é rapidamente entregue para a *face*[1] cujo pedido foi recebido. Caso o conteúdo não esteja presente no *cache*, o roteador verifica sua tabela de interesses pendentes (*PIT - pending interest table*). Se alguma *face* solicitou o mesmo conteúdo e ainda não foi atendida, o pedido é agregado a essa entrada, adicionando a *face* pela qual o pedido foi recebido. Desta forma, os roteadores evitam que requisições para o mesmo conteúdo sejam enviadas repetidamente para a rede. Se não houver um pedido pendente para o conteúdo solicitado, uma nova entrada é criada e o roteador então consulta a sua base de informação de encaminhamento (*FIB - forwarding information base*) para rotear o pedido em direção ao provedor do conteúdo solicitado. Cada roteador no caminho em direção ao provedor do conteúdo realiza todas essas etapas. O pacote de *Dados* contendo o conteúdo retorna pelo mesmo caminho em que foi solicitado, consumindo as entradas da PIT nos roteadores envolvidos.

A Figura 2 ilustra o funcionamento básico da arquitetura NDN, contendo um provedor de conteúdo, $P1$, quatro roteadores, $R1, R2, R3$ e $R4$ e dois usuários, $U1$ e $U2$. $P1$ possui dois conteúdos, A e B, que deseja disponibilizar para seus usuários. Na Figura 2(a), o usuário $U1$ envia um pacote de *Interesse* para solicitar o conteúdo A para o roteador $R4$, que roteia o pedido em direção ao provedor de conteúdo seguindo a rota $R4 \rightarrow R2 \rightarrow R1$. O pacote de *Dados* contendo o conteúdo A segue o caminho contrário em direção a $U1$. Cada roteador armazena a cópia em seu *cache*, caso a cópia ainda não esteja presente. Na Figura 2(b), o usuário $U2$ solicita o mesmo conteúdo A para $R4$ e como $R4$ tem o conteúdo em seu *cache*, a requisição é prontamente atendida.

[1] A arquitetura NDN nomeia as interfaces de *faces* para representar tanto interfaces de redes quanto interfaces de aplicações.

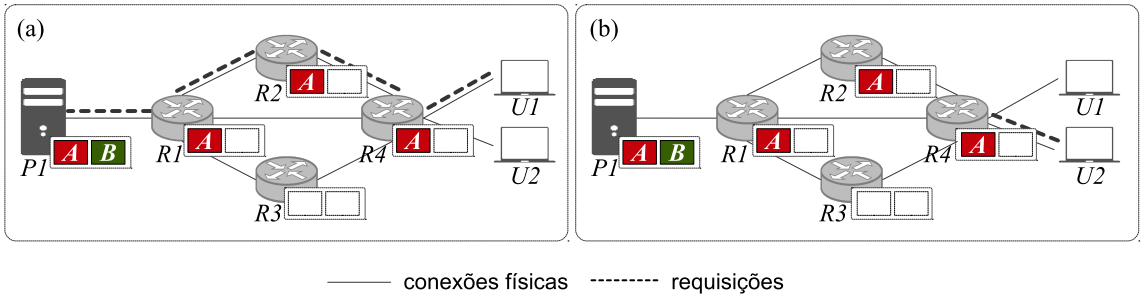


Figura 2. Infraestrutura da arquitetura NDN

4.2. Modelo de ameaças

Assume-se que os provedores de conteúdo exigem que os usuários sejam devidamente registrados na aplicação para ter acesso ao conteúdo protegido. Além disso, o provedor de conteúdo deve validar o usuário (verificar sua identidade e chave pública com uma infraestrutura de chave pública, por exemplo) para certificar-se que o usuário é de fato legítimo para o serviço e garantir seu acesso de acordo com as políticas de uso (tipo de usuário, inscrição, idade). Também se assume que o provedor de conteúdo se comporta corretamente, ou seja, não distribui conteúdo protegido ou direitos de acesso a usuários não autorizados. Os roteadores seguem o comportamento do modelo *honesto porém curioso*, em que eles desempenham corretamente suas funções (roteamento, encaminhamento e armazenamento de conteúdo em *cache*), porém podem ser curiosos e tentar acessar o conteúdo que estão roteando.

Considera-se que entidades maliciosas ($\mathcal{A}$) são usuários ilegítimos que não têm acesso ao conteúdo do provedor, ou ainda usuários legítimos que tentam acessar conteúdo ao qual não têm autorização. A intenção dessas entidades maliciosas é obter acesso ao conteúdo protegido sem ter as obrigações de usuários autorizados, tais como pagamento, verificação de dados pessoais ou ainda tipos de acesso diferenciados, como contas básicas e avançadas. Eles podem explorar o conteúdo protegido na rede pelas seguintes formas:

- aprender/descobrir o nome do conteúdo e requisitá-lo na rede;
- espionar os canais de comunicação de usuários ou interferir em pontos de acesso;
- examinar ou sondar *caches* próximos ou acessar diretamente o seu próprio *cache*.

Além disso, da mesma forma que podem solicitar conteúdo protegido na rede, as entidades maliciosas também podem recuperar as chaves de reencriptação (Seção 3) a partir dos *caches*. Também se assume que a ligação entre o nome do conteúdo e o conteúdo é devidamente encriptada e que os usuários são capazes de verificar a integridade e a autenticidade do conteúdo [Smetters and Jacobson 2009]. Por fim, assume-se também que os usuários têm acesso ao conteúdo oferecido pelo provedor de conteúdo através de uma aplicação específica e, desta forma, não há necessidade de descobrir o nome do conteúdo de antemão ou por meios não confiáveis.

4.3. Solução proposta

A solução proposta está dividida em três domínios: *domínio do provedor de conteúdo*, *domínio da rede* e *domínio do usuário*. O domínio do provedor de conteúdo engloba a encriptação do conteúdo e a geração de chaves de reencriptação para os usuários. O

domínio da rede refere-se ao roteamento e ao encaminhamento de conteúdo na rede seguindo o paradigma de ICN (Seção 4.1). O domínio do usuário é composto pela aplicação do provedor de conteúdo e pelas operações de reencriptação e decriptação do conteúdo. Tanto o provedor de conteúdo quanto uma infraestrutura de chave pública especializada podem ser responsáveis por distribuir pares de chaves pública-privada aos provedores de conteúdo e aos usuários, seguindo as especificações do algoritmo CONFIGURAÇÃO. Cada par de chave pública-privada é composto por duas chaves públicas e duas chaves privadas; essa característica é introduzida por [Chow et al. 2010] para garantir que a chave privada da fonte não seja descoberta em caso de conluio do *proxy* com o usuário no esquema original. Essa característica é extremamente importante na solução proposta, já que se considera a transferência da função do *proxy* com o usuário. A partir da aplicação disponibilizada pelo provedor de conteúdo, os usuários são autenticados e podem navegar e requisitar conteúdos, podendo ser atendidos tanto pelo provedor de conteúdo quanto por *caches* mais próximos. A seguir, são detalhadas as ações realizadas pelo provedor de conteúdo e pelos usuários para acessar um conteúdo protegido na ICN.

Encriptação e distribuição de conteúdo: o provedor de conteúdo possui um conjunto $\mathcal{C} = \{A, B, \ldots, Z\}$ de conteúdos que deseja disponibilizar. Cada conteúdo em $\mathcal{C}$ é segmentado em *chunks* e cada *chunk* é individualmente encriptado com a chave pública kp do conteúdo (todos os *chunks* pertencentes ao mesmo conteúdo são encriptados com a mesma chave). A chave privada correspondente (kv) é guardada em segredo pelo provedor de conteúdo, como de costume. A ligação entre o nome do conteúdo e o conteúdo é realizada pelo provedor com seu par de chaves pública-privada, $kp(P)$ e $kv(P)$, como sugerido em [Smetters and Jacobson 2009]. O conteúdo é distribuído conforme as requisições dos usuários, sendo armazenado em *cache* na rede de acordo com as políticas de *cache* adotadas. Neste estágio, o conteúdo pode estar em qualquer lugar na rede, porém, como a chave privada correspondente ao conteúdo é conhecida somente pelo provedor, nenhum usuário é capaz de decifrá-lo. A Figura 3(a) detalha o funcionamento dessas operações, em que o provedor possui um conjunto de conteúdos, A, B, C e D e os encripta com os seus respectivos pares de chaves, formando o conjunto $\{A\}_{kp(A)}$, $\{B\}_{kp(B)}$, $\{C\}_{kp(C)}$ e $\{D\}_{kp(D)}$, que é disponibilizado para os usuários.

Geração e distribuição da chave de reencriptação: para um usuário decriptar um conteúdo, A por exemplo, este deve ser reencriptado. Para isso, o usuário legítimo U que deseja decriptar A deve solicitar uma chave de reencriptação $rk_{A \to U}$ para o provedor. Para tal, U envia um pacote de *Interesse* para o provedor de conteúdo, requisitando uma chave de reencriptação para o conteúdo A. O provedor de conteúdo verifica se o usuário é autorizado a acessar o conteúdo A e então calcula a chave de reencriptação $rk_{A \to U}$, com base na chave pública do usuário, $kp(U)$, e na chave privada utilizada para encriptar o conteúdo, $kv(A)$, e envia para o usuário um pacote de *Dados* contendo a chave de reencriptação. A Figura 3(b) detalha essa operação. Somente o usuário U é capaz de decriptar o conteúdo A utilizando $rk_{A \to U}$, já que é necessário o uso da chave privada do usuário U (ao menos que o usuário U também divulgue sua chave privada). É inútil para uma entidade maliciosa em potencial requisitar o conteúdo e interceptar uma chave de reencriptação: ela pode ser capaz de reencriptar o conteúdo, mas a mensagem encriptada resultante só poderá ser decriptada pelo usuário que possuir a chave privada correspondente à chave de reencriptação.

Reencriptação e decriptação do conteúdo: após receber o conteúdo A e a chave de

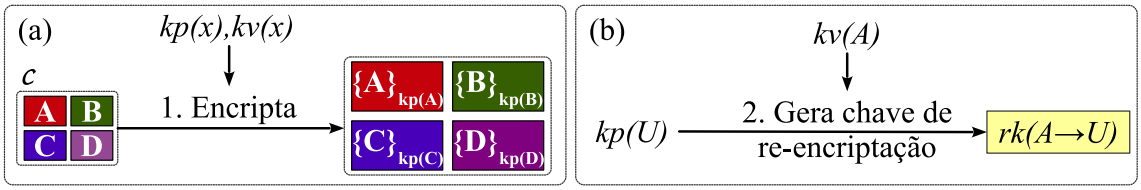


Figura 3. Domínio da fonte: (a) encriptação e (b) geração de chave de reencriptação

reencriptação $rk_{A \to U}$, o usuário U pode decriptar o conteúdo. A Figura 4(a) detalha a operação de reencriptação realizada pelo usuário U. Primeiramente, é necessário re-encriptar o conteúdo A com a chave de reencriptação correspondente. A saída deste procedimento é uma mensagem $\{A\}_{kp(U)}$, encriptada com a chave pública do usuário U, $kp(U)$. Então, o conteúdo $\{A\}_{kp(U)}$ pode ser decriptado com a chave privada do usuário U, $kv(U)$, recuperando o conteúdo A que pode ser consumido pela aplicação, conforme ilustra a Figura 4(b). As chaves de reencriptação são exclusivas de cada usuário e de cada conteúdo, portanto, cada usuário deve requisitar sua chave para o provedor de conteúdo. Desta forma, o provedor de conteúdo pode negar o envio de chaves de reencriptação para usuários que não cumpram os requisitos impostos.

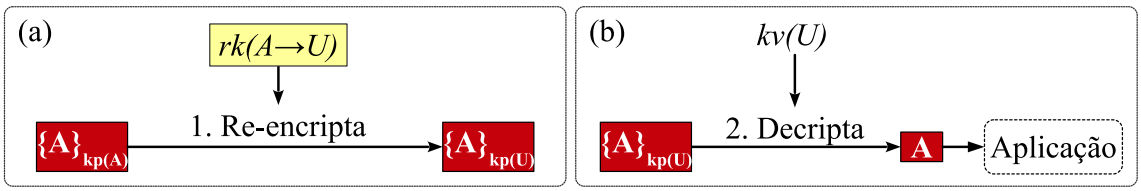


Figura 4. Domínio do usuário: (a) reencriptação e (b) decriptação

Invalidação da chave de reencriptação: uma vez que o usuário U possua a chave de reencriptação $rk_{A \to U}$ para o conteúdo A, ele é capaz de decriptar o conteúdo A sempre que desejar. Além disso, qualquer conteúdo que tenha sido encriptado pelo provedor de conteúdo com a mesma chave pública utilizada para encriptar o conteúdo A, $kp(A), kv(A)$ pode ser decriptado por U com a chave $rk_{A \to U}$. Essa é a principal razão pela qual é obrigatório que cada conteúdo tenha um par distinto de chaves pública-privada. Essa peculiaridade dificulta a invalidação das chaves de reencriptação. Ainda assim, é necessário que o provedor de conteúdo possa negar acesso aos usuários que não tenham mais permissão para acessar aos conteúdos, mesmo que tais usuários já possuam a chave de reencriptação. Uma forma de invalidar as chaves de reencriptação é renovando periodicamente a encriptação dos conteúdos com chaves públicas diferentes. Desta forma, os conteúdos teriam chaves de reencriptação correspondentes diferentes, forçando os usuários a solicitar as novas chaves de reencriptação sempre que desejarem acessar um conteúdo e suas chaves de reencriptação forem inválidas.

5. Avaliação

O objetivo da avaliação é validar o desempenho computacional do esquema de reencriptação por *proxy* proposto por [Chow et al. 2010]. Este esquema não possui uma implementação disponível, portanto, é fundamental avaliar a sua viabilidade computacional. Para isso, implementamos em Python, versão 2.7, os seis algoritmos do esquema:

CONFIGURAÇÃO, GERAÇÃO DE CHAVES, ENCRIPTAÇÃO, DECRIPTAÇÃO, GERAÇÃO DE CHAVE DE REENCRIPTAÇÃO e REENCRIPTAÇÃO[2]. A Tabela 1 descreve os parâmetros utilizados na validação[3].

Tabela 1. Parâmetros utilizados na avaliação da solução

Parâmetro	Valor	Parâmetro	Valor
Tamanho da chave (k)	1024, 2048 bits	Funções de *hash* H_1, H_3, H_4	mod q
Tamanho da mensagem (ℓ_0)	0.5, 1, 2, 4, 8, 16, 32 Kb	Função de *hash* H_2	mod $2^{(\ell_0+\ell_1)}$
Parâmetro de segurança (ℓ_1)	160 *bits*		

A validação do esquema foi realizada em um *notebook* com processador Core 2 Duo 1.66GHz, 32bits, 2Gb RAM e sistema operacional Ubuntu 13.10. A métricas adotadas foram o tempo para encriptar e gerar as chaves de reencriptação, que são ações desempenhadas pela fonte, e o tempo para reencriptar e decriptar o conteúdo, ações desempenhadas pelo usuário. A validação é realizada com diferentes tamanhos de mensagens, que representam os *chunks* enviados pela ICN. Um conteúdo é formado por um conjunto de *chunks*. Ressaltamos que, apesar do tamanho padrão de um *chunk* na arquitetura NDN ser 4Kb [Salsano et al. 2012], variamos os tamanhos dos *chunks* para ter uma noção mais clara do comportamento do esquema de reencriptação por *proxy*. Além disso, as mensagens foram encriptadas com diferentes tamanhos de chaves. Comparou-se o desempenho computacional do esquema de reencriptação por *proxy* com um outro esquema de criptografia assimétrica, o RSA. Os resultados apresentados são a média de 35 execuções do algoritmo, com um intervalo de confiança de 95%. Os algoritmos de CONFIGURAÇÃO e de GERAÇÃO DE CHAVES podem opcionalmente ser executados em uma infraestrutura de chaves públicas e, portanto, não consideramos os custos computacionais desses algoritmos. Contudo, assume-se que os provedores de conteúdos e os usuários conheçam de antemão suas respectivas chaves públicas-privadas e os parâmetros do sistema.

A Figura 5 apresenta os resultados obtidos com as operações desempenhadas pelo provedor de conteúdo: ENCRIPTAÇÃO e GERAÇÃO DE CHAVES DE REENCRIPTAÇÃO. A operação de encriptação, apresentada na Figura 5 (a), tem desempenho satisfatório e similar ao RSA, encriptando mensagens de 1 a 32Kb em menos de 200ms. Contudo, enquanto o esquema de [Chow et al. 2010] apresenta um comportamento linear com relação ao tamanho da mensagem a encriptar, o RSA apresenta um comportamento de crescimento ao aumentar o tamanho da mensagem, o que fica evidenciado no caso de mensagens de 32Kb, que o RSA tem desempenho inferior ao esquema de reencriptação por *proxy* adotado. Contudo, vale ressaltar que para pacotes de 4Kb, que é o padrão da arquitetura NDN, tanto o esquema de reencriptação por *proxy* quanto o RSA tem desempenho similar, com tempo de processamento abaixo de 200ms. Porém, deve-se considerar que o esquema de reencriptação por *proxy* possui a operação extra de geração de chaves de reencriptação na fonte, operação não presente no RSA. O tempo para gerar a chave de reencriptação também é menor que 200ms para mensagens de 4Kb. Porém, o tempo de processamento aumenta com o tamanho das mensagens, conforme ilustra a Figura 5 (b). Como o tamanho padrão de *chunks* é 4Kb, isso não representa uma grande questão. Contudo, pode ser

[2]A implementação desses algoritmos está disponível em *http://www.inf.ufpr.br/elisam/proxy*.

[3]Neste estágio, escolhemos funções de *hash* simples apenas para a validação da solução, portanto, não aferimos a segurança das mesmas.

um problema caso adote-se tamanhos de *chunks* grandes, já que no esquema proposto a fonte deve gerar chaves de reencriptação para os conteúdos a medida em que os usuários solicitam acesso. Também vale ressaltar que os tempos de processamento podem variar de acordo com o *hardware* utilizado.

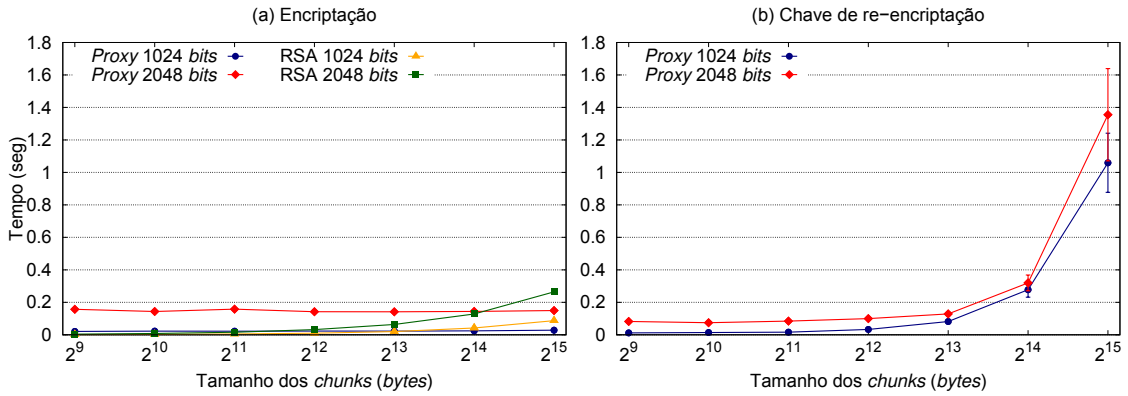


Figura 5. Avaliação da encriptação e da geração de chave de reencriptação

A Figura 6 apresenta os resultados obtidos com as operações desempenhadas pelo usuário: REENCRIPTAÇÃO e DECRIPTAÇÃO. A reencriptação constitui uma tarefa extra em relação aos esquemas tradicionais de criptografia assimétrica. De todas as operações do esquema de reencriptação por *proxy*, a reencriptação é a que apresentou a maior carga computacional, conforme ilustra a Figura 6(a). Para *chunks* de 4Kb, o tempo para processamento da reencriptação com uma chave de 2048 *bits* é de aproximadamente 800ms e aumenta conforme o tamanho das mensagens. Em contrapartida, a operação de decriptação é a operação menos custosa em termos de processamento, como apresenta a Figura 6(b). Para *chunks* de 4Kb a decriptação ocorre em aproximadamente 10ms para chaves de 1024 *bits* e 70ms para chaves de 2048 *bits*. Ainda assim o esquema de reencriptação por *proxy* se apresenta mais escalável que o RSA, que tem um alto custo computacional para decriptação, mesmo considerando *chunks* pequenos como o de 4Kb, em que o tempo para decriptação é próximo de 1 segundo (de fato, na prática, o RSA é utilizado para cifrar *hashes*). Isso ocorre mesmo que se considere a soma dos tempos das operações de reencriptação e decriptação para o esquema de reencriptação por *proxy*, conforme ilustra a Figura 6(c). A junção dessas operações é justificável, pois são operações dependentes e sempre são executadas em conjunto.

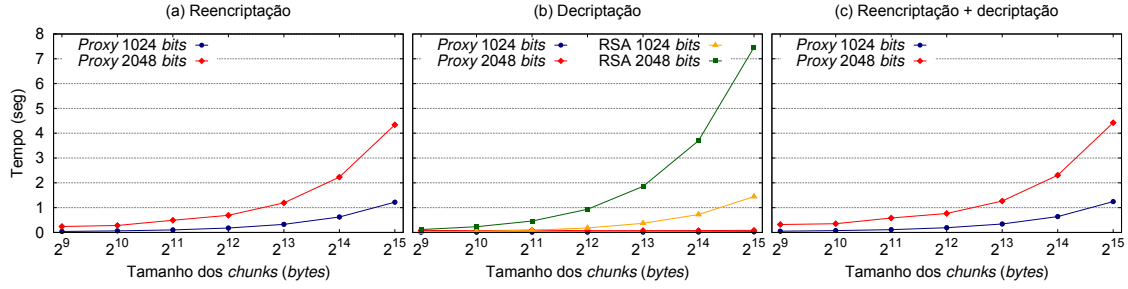


Figura 6. Avaliação da reencriptação e decriptação

6. Discussão

O principal objetivo da solução proposta é fornecer uma forma adequada para os provedores controlarem o acesso aos seus conteúdos em ICN. Os principais obstáculos para alcançar tal propriedade vêm de duas características intrínsecas do paradigma de ICN: (i) a implementação de conteúdo nomeado e (ii) o *cache* ubíquo de tais conteúdos. A seguir, apresenta-se uma discussão acerca da solução proposta com relação ao desempenho, segurança e adequação da solução sob o foco do paradigma de ICN.

Desempenho: o provedor de conteúdo precisa computar os algoritmos de encriptação e de cálculo das chaves de reencriptação. De acordo com os resultados obtidos na validação, a carga extra imposta pelo cálculo das chaves de reencriptação, em comparação com outras alternativas, não deve representar um grande impacto na questão de desempenho do provedor de conteúdo. A quantidade de chaves de reencriptação que a fonte deve calcular depende da quantidade de usuários e da quantidade de conteúdo que essa fonte disponibiliza. Contudo, a chave é criada sob demanda, especificamente para o conteúdo que o usuário deseja acessar; desta forma, a fonte não desperdiça recursos ao calcular chaves de reencriptação que não serão utilizadas. De qualquer forma, pode-se assumir que os provedores de conteúdo podem superar eventuais problemas com desempenho ao adotar estratégias de balanceamento de carga. Do ponto de vista do cliente, a questão é a tarefa extra de reencriptar o conteúdo antes de decifrá-lo, em comparação aos métodos tradicionais de criptografia assimétrica. Enquanto que essa sobrecarga é razoável para *chunks* padrão de 4Kb, ela se torna inviável com tamanhos de *chunks* maiores e possivelmente impraticável com tamanhos de chaves maiores que 2048 *bits*, inclusive para *chunks* de 4Kb. Essa sobrecarga deve ser cuidadosamente investigada, principalmente ao levar em consideração que os usuários podem utilizar dispositivos móveis com recursos escassos de memória e processamento para acessar aos conteúdos da fonte. Uma forma de melhorar os tempos de computação para a reencriptação no cliente é realizar os cálculos ao receber o primeiro *chunk* e então reutilizá-los nos *chunks* seguintes. Além disso, como a função de reencriptar está no usuário, uma investigação sobre como associar as funções de reencriptação e decriptação de forma mais otimizada é desejável e está sendo investigada. Um outro ponto importante com relação ao desempenho está relacionado à reencriptação periódica do conteúdo com chaves diferentes, para realizar a revogação de chaves de reencriptação. Uma das questões que apoiam esse processo é um dos problemas abertos em ICN apontados em [Kutscher et al. 2014]. Nesse documento, levanta-se uma preocupação com relação à robustez das chaves públicas-privadas dos provedores de conteúdos contra ataques de força bruta, já que entidades maliciosas podem recuperar um conjunto relativamente grande de conteúdo criptografado com a mesma chave. Desta forma, a reencriptação periódica dos conteúdos pode ser relevante para evitar tais ataques.

Segurança: ao utilizar um esquema de criptografia assimétrica ao invés de criptografia simétrica, como tradicionalmente proposto para controle de acesso em ICN, torna-se potencialmente mais difícil que usuários não autorizados acessem conteúdos protegidos. Por exemplo, em soluções que empregam a criptografia simétrica, é suficiente que uma entidade maliciosa divulgue o segredo para corromper o conteúdo. Na solução proposta, seria necessário que um usuário legítimo divulgasse tanto a sua chave privada como a sua chave de reencriptação. Mesmo assim, somente o conteúdo relacionado àquela chave estaria corrompido. O provedor pode, simultaneamente, implementar medidas que restringem a quantidade de aplicações concomitantes com a mesma conta de usuário, tornando ainda

menos provável que os usuários divulguem suas chaves privadas e de reencriptação, sob o risco de serem penalizados. Apesar de os esquemas de reencriptação por *proxy* tradicionalmente considerarem os *proxies* entidades confiáveis, a solução proposta se abstém de tal asserção ao eliminar a entidade *proxy* da rede e transferir as funções de reencriptação para o usuário, que a executa através da aplicação. Desta forma, os usuários não têm incentivos para agir maliciosamente ao realizar as funções que antes eram atribuídas a um *proxy*. Além disso, a solução proposta permite que os conteúdos sejam armazenados em *cache* sem restrições, inclusive na presença de usuários maliciosos que de alguma forma descobrem o conteúdo em *cache* e o solicitam. Como não possuem a chave de reencriptação, não podem acessar o conteúdo. Além disso, se o usuário malicioso tentar solicitar a chave de reencriptação para a fonte, a fonte simplesmente nega o pedido, fazendo com que o usuário malicioso tenha o conteúdo mas não consiga acessá-lo.

Adequação à ICN: um dos principais objetivos da solução proposta é a adequação ao paradigma de ICN. Neste sentido, a solução não implica mudanças na arquitetura de ICN, já que somente os provedores de conteúdo e os usuários estão envolvidos nas ações de encriptação e decriptação do conteúdo. Como a rede não é carregada com requisitos específicos, ela fica livre para rotear e encaminhar os pacotes para quem quer que requisite, na sua melhor forma. Além disso, nenhuma função de segurança é transferida para elementos da rede: os roteadores não precisam verificar chaves ou validar políticas de acesso. Entretanto, o processo de revogação de chaves ainda implica em pelo menos uma desvantagem: por uma janela de tempo, o conteúdo antigo pode ser acessado por usuários que possuem a chave de reencriptação correspondente, caso o conteúdo antigo ainda esteja em algum *cache*. Além disso, para renovar o conteúdo disponível na rede, é necessário que o armazenamento de conteúdo nos *caches* tenha um tempo de vida limitado; de outra forma, os usuários continuam requisitando pelo conteúdo antigo e os *caches* nunca seriam renovados. De qualquer forma, o paradigma de ICN já prevê uma noção de atualidade para os conteúdos em *cache*, em que a fonte pode substituir por um conteúdo mais recente. Uma outra alternativa seria incorporar um carimbo de tempo no nome do conteúdo e configurar a aplicação para que requisite o conteúdo com o carimbo de tempo apropriado. De qualquer forma, a questão da revogação de chaves necessita de uma investigação mais profunda.

7. Conclusão

Este trabalho propôs uma solução de controle de acesso baseada em reencriptação por *proxy* que permite que somente usuários autorizados possam acessar conteúdos em uma arquitetura de ICN, mesmo na presença de entidades maliciosas. Além disso, a solução proposta garante os benefícios do uso do *cache* e não introduz mudanças significativas nos provedores e na rede. Cada conteúdo é encriptado com uma chave pública correspondente; para acessar o conteúdo, os usuários devem requisitar uma chave de reencriptação para o provedor. Desta forma, o provedor de conteúdo tem um controle de acesso ativo para o conteúdo, permitindo ou negando a chave de reencriptação de acordo com suas políticas. Mesmo que uma entidade maliciosa recupere o conteúdo e uma chave de reencriptação, ainda assim não é possível que ela acesse o conteúdo. As simulações realizadas mostram que a solução proposta apresenta uma sobrecarga mínima nos provedores de conteúdo e nos usuários. Como trabalhos futuros, planeja-se refinar a solução para invalidação de chaves de reencriptação, além de simular a solução proposta com diferentes políticas de *cache*. Planeja-se ainda explorar a junção das funções de reencriptação e decriptação no usuário como forma de melhorar o desempenho do usuário ao decriptar um conteúdo.

Referências

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36.

Ateniese, G., Fu, K., Green, M., and Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transaction on Information System Security*, 9(1):1–30.

Brito, G. M. d., Velloso, P. B., and Moraes, I. M. (2012). *Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet*, chapter 5, pages 211–264. Minicursos do XXX Simpósio Brasileiro de Redes de Computadores de Sistemas Distribuídos.

Chow, S., Weng, J., Yang, Y., and Deng, R. (2010). Efficient unidirectional proxy re-encryption. In Bernstein, D. and Lange, T., editors, *Progress in Cryptology – AFRICACRYPT 2010*, volume 6055 of *Lecture Notes in Computer Science*, pages 316–332.

Fotiou, N., Marias, G. F., and Polyzos, G. C. (2012). Access control enforcement delegation for information-centric networking architectures. In *2nd ACM SIGCOMM Workshop on Information-centric networking (ICN '12)*, pages 85–90.

Hamdane, B., Msahli, M., Serhrouchni, A., and El Fatmi, S. (2013). Data-based access control in named data networking. In *9th International Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom '13)*, pages 531–536.

Ion, M., Zhang, J., and Schooler, E. (2013). Toward content-centric privacy in ICN: attribute-based encryption and routing. In *3rd ACM SIGCOMM Workshop on Information-centric networking (ICN '13)*, pages 39–40.

Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M., Briggs, N., and Braynard, R. (2012). Networking named content. *Communications of the ACM*, 55(1):117–124.

Kissel, Z. and Wang, J. (2013). Access control for untrusted content distribution clouds using unidirectional re-encryption. In *2013 International Conference on High Performance Computing and Simulation (HPCS)*, pages 49–56.

Kutscher, D., Pentikousis, K., Psaras, I., Corujo, D., Saucez, D., Schmidt, T., and Waehlisch, M. (2014). ICN research challenges. http://www.ietf.org/id/draft-kutscher-icnrg-challenges-02.txt. Work in progress.

Misra, S., Tourani, R., and Majd, N. E. (2013). Secure content delivery in information-centric networks: design, implementation, and analyses. In *3rd ACM SIGCOMM workshop on Information-centric networking (ICN '13)*, pages 73–78.

Papanis, J. P., Papapanagiotou, S. I., Mousas, A. S., Lioudakis, G. V., Kaklamani, D. I., and Venieris, I. S. (2013). On the use of attribute-based encryption for multimedia content protection over information-centric networks. *Transactions on Emerging Telecommunications Technologies*, pages 1–14.

Salsano, S., Detti, A., Cancellieri, M., Pomposini, M., and Blefari-Melazzi, N. (2012). Transport-layer issues in information centric networks. In *2nd Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pages 19–24. ACM.

Singh, S., Puri, A., Singh, S. S., Vaish, A., and Venkatesan, S. (2012). A trust based approach for secure access control in information centric network. *International Journal of Information and Network Security (IJINS)*, 1(2):97–104.

Smetters, D. and Jacobson, V. (2009). Securing network content. Technical report, PARC TR-2009-1.

Wood, C. and Uzun, E. (2014). Flexible end-to-end content security in ccn. In *IEEE Consumer Communications and Networking Conference*, CCNC '14, pages 1–8.

Xiong, H., Zhang, X., Zhu, W., and Yao, D. (2012). Cloudseal: End-to-end content protection in cloud-based storage and delivery services. In *Security and Privacy in Communication Networks*, volume 96, pages 491–500.

Protocolo para transferência parcial de conhecimento e sua aplicação à verificação segura de marcas d'água*

Raphael Carlos Santos Machado[2],
Davidson Rodrigo Boccardo[2],
Vinícius Gusmão Pereira de Sá[1],
Jayme Luiz Szwarcfiter[1,2]

[1]Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, RJ – Brasil

[2]Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO)
Duque de Caxias, RJ – Brasil

{rcmachado,drboccardo}@inmetro.gov.br,
vigusmao@dcc.ufrj.br, jayme@nce.ufrj.br

Abstract. *Let $y = f(x)$ for some one-way function f. We present a simple algorithm that allows that the bits of x are but partially exhibited in a demonstration, through a zero-knowledge proof scheme, that x is indeed an element of the pre-image of y under f. As an application, we show that it is possible to disclose a watermark embedded into a digital artifact without the need of revealing its location. The result is a secure verification protocol for software watermarking which does not increase the likelihood that an attacker is successful in a removal attack.*

Resumo. *Seja $y = f(x)$ para uma função one-way f. Apresentamos um algoritmo simples que permite exibir a terceiros apenas parte dos bits de x numa demonstração, por meio de um esquema de prova de conhecimento nulo, de que x pertence de fato à pré-imagem de y sob f. Como aplicação, mostramos que é possível exibir uma marca d'água embarcada em um artefato digital sem necessidade de revelar sua localização. O resultado é um protocolo seguro para verificação de marcas d'água de software que não aumenta a probabilidade de um atacante ser bem-sucedido em um ataque de remoção.*

1. Introdução

Em sua tese de doutorado, Joe Kilian [Kilian 1990] apresenta o seguinte problema. Bob deseja fatorar um número n de 500 bits que, sabe-se, é o produto de cinco números primos de 100 bits. Alice conhece um dos fatores, denotado q, e está disposta a vender 25 de seus bits a Bob. Kilian propõe um método que possibilita a Alice comprovar que ela de fato conhece um dos fatores de n, e ainda permite que tal comprovação aconteça sobre *bits individuais* de q. O protocolo proposto por Kilian não apenas permite que sejam revelados

*Trabalho parcialmente financiado por CAPES, CNPq, FAPERJ, Pronametro 52600.017257/2013 e Eletrobrás DR/069/2012.

somente alguns dos bits de q, mas utiliza esquemas de *commitment*[1] individual dos bits de q para garantir também que eles não serão revelados sem o consentimento de Alice. Finalmente, permite o emprego de *oblivious transfer* [Rabin 1981] de tal modo que a própria Alice desconheça o conjunto dos bits efetivamente revelados.

No presente trabalho, apresentamos um protocolo para um cenário simplificado de revelação de alguns dos bits de q, permitindo observar aspectos essenciais do que denominamos "transferência parcial de conhecimento". No cenário proposto, Alice não possui interesse financeiro sobre os bits a serem transferidos: Alice está disposta a revelar alguns dos bits de q a quem quer que deseje conhecê-los. Por outro lado, Alice somente concorda em revelar um determinado subconjunto dos bits de q — por convenção, assumiremos que Alice sempre revela os bits mais significativos de q, embora essa escolha seja arbitrária. O fato de que tal conjunto é pré-determinado dispensa o uso de *oblivious transfer*.

Pelo protocolo proposto, Alice é capaz de exibir os bits mais significativos de q, comprovando a quem possa interessar que, de fato, tratam-se de parte dos bits de um dos fatores de n. O protocolo é simples e intuitivo, e faz uso de reduções polinomiais e de provas de conhecimento nulo, baseando-se na Hipótese da Dificuldade da Fatoração (HDF). É fácil verificar que as técnicas propostas podem ser adaptadas a problemas clássicos notadamente difíceis, tais como logaritmo discreto.

Como aplicação do protocolo proposto, apresentamos um cenário de verificação segura de marcas d'água. A principal vantagem do uso de um protocolo de transferência parcial de conhecimento é a possibilidade de divulgar informações de autoria e propriedade, armazenadas exatamente nos bits a serem exibidos, sem que a necessidade de verificação eventual da marca d'água torne mais fácil sua remoção por parte de um atacante. Como a transferência de conhecimento é parcial, o atacante interessado em remover a marca d'água não disporá de informação suficiente para localizá-la dentro do artefato em que está embarcada, de modo que sua remoção permanecerá tão difícil após a verificação quanto antes dela.

O artigo está organizado da seguinte forma. Na Seção 2, apresentamos um protocolo que permite demonstrar que um conjunto de bits é o prefixo de um dos fatores de um número, sem que seja necessário, no entanto, exibir qualquer dos fatores desse número. Na Seção 3, descrevemos objetivos e conceitos básicos associados a marcas d'água digitais. Na Seção 4, mostramos como o protocolo de transmissão parcial de conhecimento pode ser adaptado à construção de um esquema de marcas d'água digitais que é resistente a ataques de remoção mesmo após a verificação de uma marca d'água embarcada em um artefato digital. Na Seção 5, apresentamos trabalhos relacionados, e, na Seção 6, nossas considerações finais.

2. Protocolo para transferência parcial de conhecimento

Dado um inteiro positivo n que é o produto de dois números primos p e q, queremos ser capazes de mostrar que uma dada sequência de bits k corresponde aos bits mais significativos (ou *prefixo*) de p, sem revelar quaisquer dos fatores. O protocolo proposto é baseado, essencialmente, na aplicação de esquemas de prova de conhecimento nulo e

[1]Optamos por manter, neste texto, termos originais em inglês cujos equivalentes em língua portuguesa não se encontram ainda bem estabelecidos, como *commitment* de bits (e bits *committed*) e *oblivious transfer*.

em transformações polinomiais entre variantes do problema da fatoração de um inteiro e variantes do problema da satisfabilidade boolena.

2.1. Reduzindo EQUICOMPOSITE a SAT

Inicialmente, mostramos que o problema de determinar se um número é composto pode ser facilmente reduzido ao problema de se determinar se uma expressão lógica é satisfatível, problema esse conhecido como SAT [Schaefer 1978]. Mais precisamente, consideramos a variante EQUICOMPOSITE do problema de fatoração, em que se quer determinar se um inteiro n pode ser escrito como o produto de dois fatores, cada um dos quais com no máximo $\lceil log_2(n)/2 \rceil$ bits.

EQUICOMPOSITE
Entrada: número binário n, com $\lceil log_2(n) \rceil$ bits.
Saída: SIM, se n é o produto de dois números de bitsize até $\lceil log_2(n)/2 \rceil$;
NÃO, caso contrário.

Para tratar o problema EQUICOMPOSITE por meio de provas de conhecimento nulo, estudaremos a implementação de variantes da operação de multiplicação por meio de circuitos combinacionais, ou, equivalentemente, por meio de expressões lógicas envolvendo os bits dos operandos.

Produto de inteiros como uma função lógica

É sabido que a operação de produto de dois números binários pode ser descrita na forma de um circuito combinacional, de tal forma que cada dígito do resultado é uma expressão lógica sobre os dígitos dos operandos. Por questão de completude, revisamos brevemente a teoria sobre como construir um multiplicador binário.

Somando bits. É fácil implementar um circuito combinacional simples que recebe como entrada dois bits A e B (os operandos) e um terceiro bit C_i, o "vai um" (gerado por um somador em estágio anterior), e que retorna como saída o bit S resultante da soma dos três bits recebidos e um novo bit C_o de "vai um" (Figura 1).

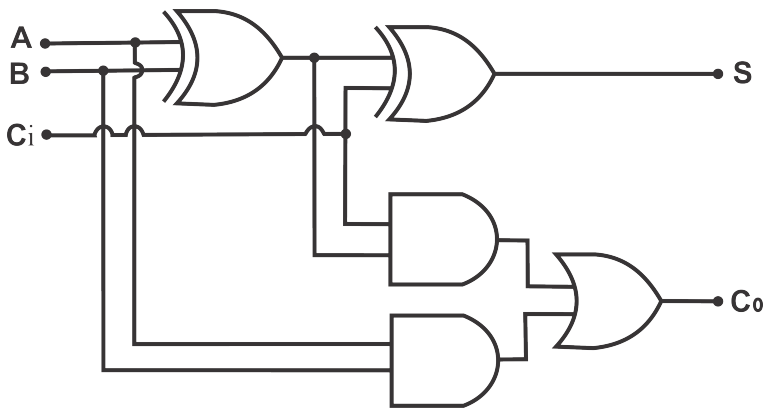


Figura 1. Somador completo

Observe que tanto o bit S quanto o bit C_o podem ser descritos como uma expressão lógica aplicada sobre os bits A, B e C_i:

- $S = (A \oplus B) \oplus C_i$
- $C_o = (A \cdot B) + (C_i \cdot (A \oplus B))$

Naturalmente, o XOR ("ou exclusivo", representado por $\oplus$) pode ser substituído por operações OR ($+$) e AND ($\cdot$), de acordo com a fórmula $A \oplus B = \bar{A}B + A\bar{B}$.

Encadeando somadores. Para efetuar a soma de números binários com mais de um bit, basta encadear somadores completos, sempre enviando o "vai um" de saída de um estágio à entrada do estágio seguinte (Figura 2). Mais uma vez, cada um dos bits de saída

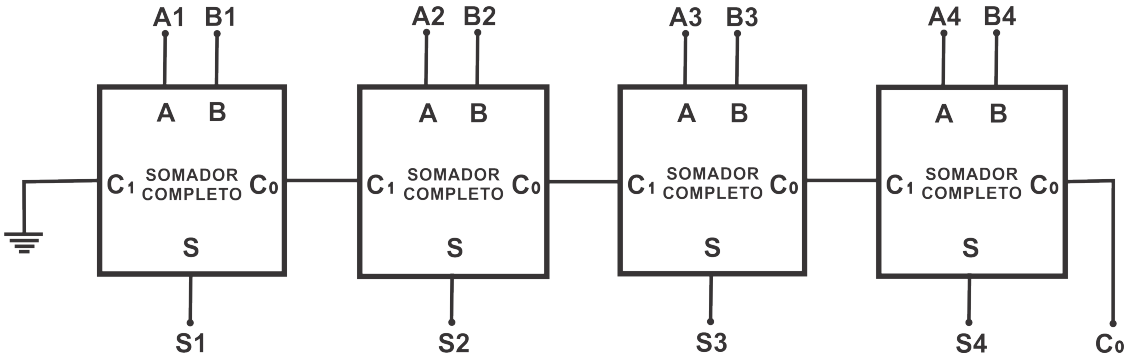


Figura 2. Somador de quatro bits

pode ser descrito como uma expressão lógica aplicada sobre os bits de entrada.

Multiplicando por potências de dois. A multiplicação por dois, em binário, pode ser executada como um simples deslocamento à esquerda, com a inclusão de um bit zero como dígito menos significativo da saída. Denotaremos o deslocamento à esquerda em i bits (multiplicação por 2^i) de um número binário B por $B << i$.

Obtendo o produto de dois números binários. De maneira simplificada, a operação de multiplicação sobre binários pode ser entendida como uma série de adições e multiplicações por dois. Por exemplo, para multiplicar $A = A_3A_2A_1A_0$ por $B = B_3B_2B_1B_0$, começamos pelo bit mais à direita de um dos operandos, digamos, A. Se o bit A_0 é igual a 1, então, adicionamos o valor do outro operando, B, ao resultado C (que é inicialmente zero); se o bit A_0 é igual a 0, então nenhum valor é adicionado. Para cada um dos bits consecutivos A_i de A, se e somente se $A_i = 1$, efetuamos um deslocamento à esquerda de tamanho i em B (ou seja, multiplicamos B por 2^i) e o somamos ao resultado. O resultado, escrito como uma expressão lógica, equivale a $C = B \wedge A_0 + (B << 1) \wedge A_1 + (B << 2) \wedge A_2 + (B << 3) \wedge A_3$ (Figura 3).

Construindo uma saída única. Sabendo descrever o produto de dois números binários na forma de um circuito combinacional, é fácil adaptá-lo para um circuito modificado que possui um único bit de saída cujo valor é 1 se e somente se um determinado número n é equicomposto. Para isso, basta adicionar portas NOT a cada saída do circuito multiplicador associada a um bit de n que deve ser 0, e conectar todas as saídas a uma única porta AND.

Formalmente, um circuito UNI-MULT(d, n), onde d é um inteiro e n é um número binário, é construído da forma esquematizada na Figura 4, com um circuito multiplicador

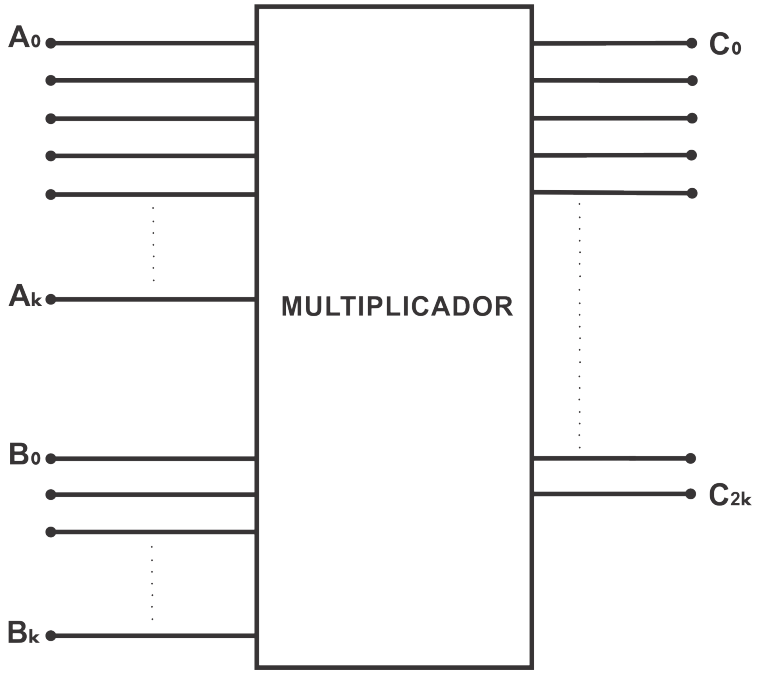


Figura 3. Multiplicação

de dois números binários de d bits, com uma porta NOT após cada saída do multiplicador associada a um bit 0 de n, e com uma porta AND conectando todas as $2d$ saídas (invertidas ou não). O Teorema 1, cuja prova é imediata, resume o resultado desejado.

Teorema 1 *O circuito UNI-MULT*(d, n) *retorna o bit* 1 *se e somente se o número binário* n *puder ser escrito como o produto de dois números binários de (até)* d *bits.*

2.2. O problema PREFATOR

Consideramos, agora, o problema de se determinar se um número pode ser escrito como o produto de dois outros números, um dos quais possui um conjunto de bits cujos valores são previamente fixados. Mais precisamente, considere o seguinte problema de decisão, ao qual denominamos PREFATOR.

PREFATOR
Entrada: números binários n e k.
Saída: SIM, se n é equicomposto e possui um fator do qual k é prefixo;
NÃO, caso contrário.

Sabendo-se reduzir EQUICOMPOSITE a SAT, torna-se simples entender como reduzir o problema PREFATOR a SAT. De fato, nosso objetivo é determinar se um número é o produto de dois outros números, um dos quais começando por um conjunto de bits pré-fixados. Nossa estratégia é construir um circuito semelhante ao da Figura 4, mas com a "transferência" de alguns bits da entrada diretamente para o último estágio do circuito, que recebe uma porta AND adicional conforme o diagrama da Figura 5.

Formalmente, um circuito PRE-MULT(d, n, k), onde d é inteiro e n e k são números binários, é construído da forma esquematizada na Figura 5. Inicialmente, toma-

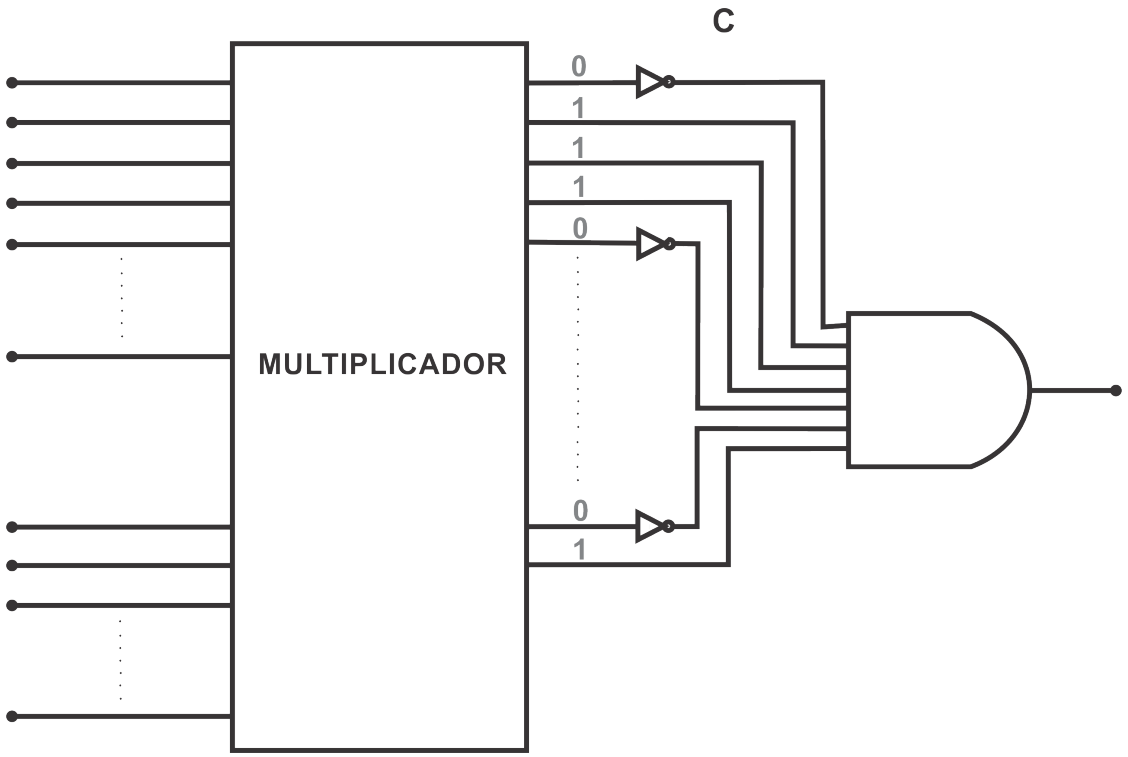


Figura 4. UNI-MULT: fixando os bits da saída com um AND final

se um circuito UNI-MULT(d, n). Para cada bit de entrada do circuito UNI-MULT(d, n) associado a um bit de k, cria-se uma derivação, que será conectada a uma porta NOT se esse bit for 0 em k. As derivações são todas conectadas a uma porta AND, assim como o bit de saída do circuito UNI-MULT(d, n). O Teorema 2 resume o que o circuito PRE-MULT permite responder.

Teorema 2 *O circuito PRE-MULT*(d, n, k) *retorna o bit* 1 *se e somente se o número binário* n *puder ser escrito como o produto de dois números binários de até* d *bits, um dos quais possuindo* k *como prefixo.*

2.3. Convertendo para a forma normal conjuntiva

O leitor observará que, novamente, a saída do circuito PRE-MULT(d, n, k) é uma função lógica sobre os bits de entrada. No entanto, para utilizarmos o arcabouço da teoria da complexidade computacional e suas reduções polinomiais, é necessário dispor de uma expressão lógica na forma normal conjuntiva. Felizmente, as transformações de Tseitin [Tseitin 1983] permitem construir, a partir de uma expressão lógica σ qualquer, uma nova expressão lógica σ' cujo tamanho é linear no tamanho de σ. Além disso, a transformação é executada em tempo linear no tamanho de σ.

2.4. Utilizando provas de conhecimento nulo

Sabendo reduzir o problema PREFATOR a SAT, podemos de forma simples recorrer a provas de conhecimento nulo por meio de reduções polinomiais. Podemos, por exemplo, reduzir a instância SAT a uma instância de 3-COLORAÇÃO em tempo polinomial [Karp 1972], para, então, utilizar um esquema clássico de provas de conhecimento nulo para este último problema [Goldwasser et al. 1985].

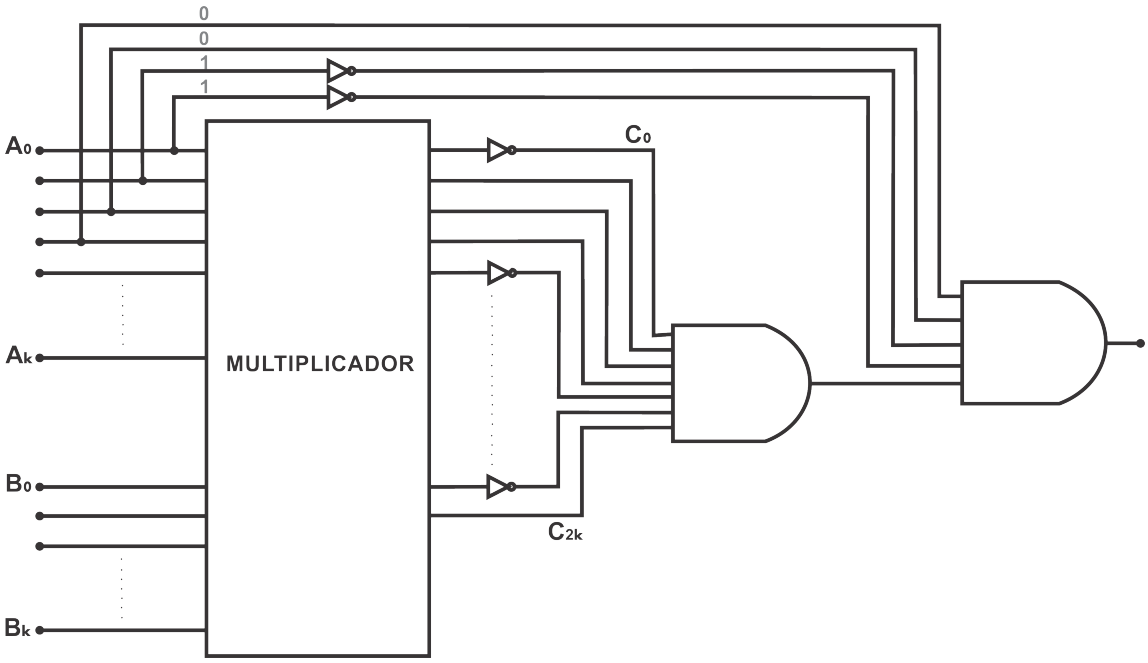


Figura 5. PRE-MULT: fixando os quatro primeiro bits de A em "1100"

3. Marcas d'água

A partir deste ponto, mostraremos como o protocolo de divulgação parcial de conhecimento que estamos propondo pode ser utilizado no contexto de um esquema de marcas d'água digitais, com o fim de demonstrar autoria ou propriedade de um artefato digital. De fato, esta é a motivação original para a construção dos protocolos de divulgação parcial de conhecimento apresentados neste artigo.

Esquemas de marca d'água são técnicas que permitem embarcar uma informação, muitas vezes de forma oculta, em um artefato qualquer, de tal maneira que, uma vez inserida, torna-se inviável remover tal informação por um agente mal intencionado. Estas informações embarcadas são utilizadas, em geral, para identificar autoria ou propriedade, ou ainda para conferir identidade única a um artefato. Com o advento do computador, tornou-se desejável embarcar informações em artefatos digitais, tais como arquivos de texto, imagem e áudio, além de programas de computador, entre outros.

Marcas d'água baseiam-se na imersão de uma informação — denominada *conteúdo* — em um artefato digital — denominado *hospedeiro* — de tal forma que o artefato digital obtido — ao qual denominamos *produto* — é, para todos os efeitos, equivalente ao hospedeiro.

Essencialmente, um esquema de marcas d'água é definido por meio de dois algoritmos. O primeiro deles, denominado *embarcador*, é um algoritmo que recebe como entrada um hospedeiro u e um conteúdo m a ser embarcado, e retorna como saída um produto $\tilde{u}$ equivalente ao hospedeiro, mas que contém embarcada a informação m. Adicionalmente, precisamos de um algoritmo *extrator* que recebe como entrada um artefato digital (produto) $\tilde{u}$, o qual possui embarcada uma informação-conteúdo m, e retorna como saída essa informação m.

As propriedades essenciais de uma marca d'água são geralmente furtividade, re-

siliência e verificabilidade. Furtividade diz respeito à propriedade de um artefato com marca d'água ser indistinguível de um artefato sem marca d'água. Resiliência diz respeito à propriedade de uma marca d'água ser de difícil remoção ou modificação, sob pena de comprometer a própria funcionalidade do artefato. Verificabilidade diz respeito à propriedade de uma marca d'água poder ser exibida a terceiros, de modo a comprovar a autoria ou a propriedade de um artefato.

A maior dificuldade em construir um esquema de marca d'água para artefatos digitais advém do fato de que tanto a informação embarcada quanto o artefato hospedeiro são, em última análise, sequências de bits, portanto, facilmente manipuláveis. Desta forma, a dificuldade em remover uma marca d'água tem se baseado na dificuldade em localizar a marca d'água [Bento et al. 2013]. Tal estratégia, no entanto, cria dificuldades quando é necessário exibir uma marca d'água, por exemplo, para comprovar a autoria de uma obra: uma vez que se conhece a localização da marca d'água, torna-se fácil removê-la.

Embora possa haver interesse em marcas d'água que necessitem ser exibidas somente uma vez — protocolos de fingerprinting são um exemplo, em que cada marca d'água identifica um único artefato, sendo embarcada somente naquele artefato, e bastando uma única verificação da marca d'água, em tribunal, para imputar culpa a quem houver feito utilização indevida do produto —, é desejável que seja possível exibir uma marca d'água sem que isto comprometa sua posterior resiliência a ataques. Denominamos *verificável* um esquema de marcas d'água que permita a verificação de uma marca d'água sem que isto reduza a dificuldade em removê-la. Note que a propriedade de verificabilidade está associada não somente à concepção da marca d'água, mas a forma de utilizá-la (verificá-la) através de um protocolo.

O problema é que o requisito de verificabilidade da marca d'água parece, de certa forma, incompatível com o requisito de resiliência: uma vez que se revela a localização da marca d'água no momento de sua verificação, desfaz-se a dificuldade, antes existente, de desabilitá-la por meio de ataques de remoção ou distorção. Mostramos a seguir como utilizar o protocolo proposto para transferência parcial de conhecimento de forma a exibir informações embarcadas em um artefato digital sem que seja necessário revelar sua localização.

4. Um esquema seguro de verificação de marcas d'água

4.1. Inserindo a marca d'água

Algumas das principais técnicas para se embarcar uma marca d'água em um artefato digital baseiam-se na possibilidade de se modificar imperceptivelmente o artefato, de forma que o produto final seja, para todos os efeitos, semanticamente equivalente ao original, e com a propriedade extra de permitir a obtenção de uma informação nele instalada de forma surreptícia. Vejamos alguns exemplos simples de classes de equivalência que podem ser exploradas.

Texto. Considere o campo de aplicação "língua portuguesa" e o conjunto dos textos escritos em português. Podemos dizer que dois textos são semanticamente equivalentes se são idênticos, a menos de substituições entre as palavras "entretanto" e "todavia".

Queria ir à praia; entretanto, choveu. $\equiv$ Queria ir à praia; todavia, choveu.

Uma outra possível classe de equivalência semântica é aquela em que períodos consecutivos são separados por "ponto final" ou por "ponto-e-vírgula".

Penso. Logo, existo. ≡ Penso; logo, existo.

Uma marca d'água que codifique um binário num texto suficientemente grande em língua portuguesa poderia utilizar a sequência de "entretanto" e "todavia" para codificar, respectivamente, os bits "1" e "0". Ou a contagem de ";" poderia corresponder ao inteiro que se deseja embarcar, ou diversas outras formas de se explorar modificações sintáticas que preservam a semântica.

Imagem digital. Considere o campo de aplicação "imagens digitais" e o conjunto dos arquivos bitmap em que cada bit é codificado como uma tripla que define a intensidade de vermelho, verde e azul (digamos, 0 a 255, cada intensidade). Uma possível classe de equivalência semântica é aquela em que intensidades ímpares podem ser reduzidas de uma unidade e intensidades pares podem ser aumentadas de uma unidade.

... (173,189,12) ... ≡ ... (172,189,13) ...

Uma marca digital poderia ser embarcada através da modificação apropriada dos códigos de cor em uma sequência de pixels iniciado em algum ponto específico da imagem. Códigos ímpares estariam, por exemplo, associados a bits "1", e códigos pares, a bits "0".

Programa de computador. Considere o campo de aplicação "programas de computador" e o conjunto dos códigos em linguagem C. Uma possível classe de equivalência semântica é aquela em que as saídas dos programas são as mesmas (desconsiderando tempo de execução ou uso de memória).

... printf("Hello World") ... ≡ ... if(1){printf("Hello World")}else{printf("Buraco")} ...

Aqui, mais uma vez, pode-se utilizar a forma (apropriadamente modificada) para embarcar o conteúdo desejado.

4.2. Verificabilidade

Nos exemplos clássicos de marca d'água embarcadas em artefatos físicos concretos, a marca d'água é visível e verificável por qualquer um que tenha acesso ao artefato que a contém. Isto é possível pois a dificuldade em remover a marca d'água é baseada no processo físico de imersão da marca d'água. Como exemplo, considere a dificuldade em remover de um documento uma marca em baixo relevo.

A necessidade de embarcar marcas d'água em artefatos digitais leva à necessidade de uma propriedade adicional à qual denominamos *verificabilidade*, que é a possibilidade de se poder atestar a presença da marca d'água sem que isto torne mais fácil, a um atacante, removê-la. Por se tratarem de artefatos digitais, torna-se relativamente fácil modificar tais artefatos. Assim, uma marca d'água concebida ingenuamente pode se tornar facilmente removível caso um atacante tenha conhecimento de sua forma e posição. Os seguintes exemplos ilustram o argumento:

Exemplo de marca d'água em texto. Considere uma marca d'água em um texto codificado na forma de uma frase cuidadosamente construída para codificar a marca d'água. Para um atacante que não tenha conhecimento da posição desta frase especial dentro do texto, a tarefa de remover a marca d'água é bastante difícil. No entanto, uma vez que o autor do texto exiba a marca d'água — revelando a frase — a remoção da marca d'água passa a ser tarefa trivial.

Exemplo de marca d'água em programa de computador. Considere uma marca d'água em um programa de computador codificada na forma de um subgrafo do grafo de fluxo de controle deste programa. Para um atacante que não tenha conhecimento da posição do subgrafo dentro do grafo do programa, a tarefa de remover a marca d'água é bastante difícil, ainda que o atacante conheça o subgrafo que a codifica, pois estaria diante de um problema difícil (e clássico) em Teoria dos Grafos, qual seja o do isomorfismo de subgrafos. No entanto, uma vez que o agente embarcador da marca d'água a exiba, revelando sua localização, a remoção da marca d'água passa a ser tarefa fácil.

Para poder demonstrar a autoria ou a propriedade de um artefato digital com base no algoritmo descrito na Seção 2, utilizaremos a seguinte estratégia. Primeiramente, codificaremos a informação de autoria ou propriedade na forma de um número binário k. Selecionamos dois números primos p e q, um dos quais possui exatamente k como seu conjunto de bits mais significativos (prefixo), e computamos o produto n dos números p e q. O produto resultante será imerso no artefato digital a ser marcado, nele aparecendo na forma de uma *substring*, isto é, subsequência de bits (mais precisamente, aparecendo como uma subsequência da sequência de bits obtida a partir do artefato digital por meio da execução do algoritmo extrator). A motivação para esta estratégia é o fato de podermos exibir k sem ser necessário revelar a localização de n.

Consideraremos agora uma pequena variação da definição do algoritmo extrator, ao qual denominamos algoritmo *pré-extrator*. O algoritmo pré-extrator, ao invés de retornar exatamente a marca d'água (previamente embarcada por meio do algoritmo embarcador), retorna uma sequência de bits — possivelmente longa — que contém a marca d'água como substring. Mais precisamente, a substring em questão será, como vimos, o produto de dois números primos, um dos quais possuindo a marca d'água como prefixo.

4.3. O problema SUBSTRING-PREFATOR

Considere o seguinte problema de decisão.

SUBSTRING-PREFATOR
Entrada: números binários d e k, inteiro N.
Saída: SIM, se existe substring n de d, com N bits, tal que n é equicomposto e um de seus fatores possui k como prefixo;
NÃO, caso contrário.

É fácil ver que o problema SUBSTRING-PREFATOR também pode ser reduzido a SAT. De fato, basta construir um circuito PRE-MULT (idêntico ao construído para o problema PREFATOR) para cada substring de tamanho N, e aplicar um grande OR às saídas de cada um dos $bitsize(d) - N + 1$ circuitos.

4.4. Gerando a marca d'água

A geração da marca d'água a ser embarcada é um processo simples. Dada uma informação m a ser embarcada, basta gerar dois números primos aleatórios p e q de mesmo bitsize, de tal forma que m é prefixo de p, e computar o produto $n = p \cdot q$, sequência de bits que será, de fato, embarcada no artefato digital. A geração de q segue os métodos tradicionais de sorteio de número aleatório seguido de teste de primalidade (por exemplo, Miller-Rabin) até que se obtenha um número primo. A geração de p segue uma abordagem ligeiramente modificada: sorteia-se um número aleatório, mas concatena-se o número sorteado à direita de m para, na sequência, testar-se a primalidade do número obtido — e o processo é repetido até obter-se um número primo.

4.5. Inserindo a marca d'água

O processo de inserção de marca d'água tem por objetivo modificar o artefato digital a ser marcado, fazendo com que a sequência de bits $n = p \cdot q$ apareça como uma substring da string retornada pelo programa extrator. Na prática, o processo exato de inserção — e da consequente extração — irá depender do tipo de artefato digital a ser marcado. No caso de um arquivo texto, por exemplo, o processo irá depender apenas do esquema de codificação utilizado para embarcar informações no texto. Na Seção 3, por exemplo, descrevermos um esquema em que a sequência de palavras "todavia" e "entretanto" determinaria uma informação codificada. Neste caso, bastaria selecionar uma sequência destas palavras, substituindo-as apropriadamente para conter os bits construídos pelo protocolo. Exemplos análogos poderiam ser construídos para outros campos de aplicação, e a especificação detalhada de cada caso está fora do escopo do presente trabalho.

4.6. Verificando a marca d'água

O processo de verificação da marca d'água contempla as seguintes etapas:

1. Extração da sequência de bits associada ao artefato digital — sequência esta que pode ser muito longa, mas que contém, como substring, $n = p \cdot q$.
2. Transformação da sequência gerada em uma instância de SAT, e daí para uma instância de 3-COLORAÇÃO.
3. Utilização de esquema de prova de conhecimento nulo para demonstrar que o grafo obtido no passo anterior é, de fato, 3-colorível.

A extração da sequência de bits a que nos referimos no passo inicial pode ser feita por meio de algoritmos específicos para esse fim, os quais devem ser definidos para cada campo de aplicação e seus correspondentes artefatos digitais. A transformação para uma instância de SAT é exatamente o algoritmo descrito na Seção 2, enquanto a redução polinomial de SAT para 3-COLORAÇÃO é um resultado clássico da literatura [Karp 1972]. Um esquema de prova interativa de conhecimento nulo para 3-COLORAÇÃO é descrito em [Goldwasser et al. 1985] e pode ser utilizado no último passo para exibir a marca d'água sem apresentar n ou quaisquer de seus fatores.

5. Trabalhos relacionados

Marcas d'água em artefatos digitais têm sido uma área ativa de pesquisa desde a década de 90, com uma grande quantidade de trabalhos e diferentes técnicas para embarcar marcas d'água em áudio, vídeo, imagem e software, como sumariadas em

[Collberg and Nagra 2009]. Técnicas de marca d'água podem ser classificadas com base nas técnicas de embarcação e extração, e em sua natureza estática ou dinâmica. No entanto, marcas d'água dinâmicas são específicas para software. Tipicamente, marcas d'água em software são embarcadas por meio de substituição de código, reordenação de código, alocação de registros, grafos estáticos ou dinâmicos, interpretação abstrata e utilização de predicados opacos. Um detalhamento dessas técnicas pode ser encontrado nos trabalhos de [Zhu et al. 2005, Hamilton and Danicic 2011].

Uma marca d'água de software é estática caso a marca d'água esteja contida no segmento de código ou de dados do software; ou dinâmica, caso esteja em algum estado de execução do software, ou seja, construída durante a execução do programa. Marcas d'água dinâmicas apresentam vantagens de furtividade e resiliência em relação a marcas d'água estáticas [Collberg and Nagra 2009]. Uma marca d'água dinâmica é construída durante a execução do software por meio de instruções intercaladas a instruções da aplicação, fazendo com que a codificação e a localização da marca seja desconhecida para um adversário. Isso adiciona furtividade à marca d'água, uma vez que sua localização é distinta a cada execução do programa. Isto aumenta, consequentemente, sua resiliência a ataques, uma vez que dificulta sua detecção e adulteração/remoção.

Diversos trabalhos discutem a aplicação de provas de conhecimento nulo para áudio, vídeo, imagem [Craver 1999, Adelsbach et al. 2003, Adelsbach and Sadeghi 2001]. Para nosso conhecimento, existe apenas um trabalho que aplica prova de conhecimento nulo para marcas d'água de software [Venkatachalam 2005]. A proposta de Venkatachalam, no entanto, evita que partes da marca d'água sejam reveladas durante um processo de verificação, enquanto o presente trabalho evita a apresentação de *qualquer parte* da marca d'água. Como resultado, temos um protocolo de verificação mais robusto diante de ataques de remoção. Adicionalmente, cabe destacar que o trabalho de [Venkatachalam 2005] baseia-se em um esquema clássico de prova de conhecimento nulo para o problema dos resíduos quadráticos; nosso trabalho, por outro lado, por meio de reduções polinomiais de circuitos para problemas lógicos, cria um arcabouço que permite a aplicação de qualquer problema computacionalmente difícil ao problema de verificação segura de marca d'águas.

6. Considerações finais

No presente trabalho, apresentamos um protocolo simplificado que busca responder a uma pergunta clássica de Joe Kilian [Kilian 1990] sobre a possibilidade de um transferência parcial de conhecimento a respeito da solução de um problema. O protocolo faz uso de ferramentas clássicas tais como provas de conhecimento nulo e reduções polinomiais. Concluímos o trabalho com uma série de observações a respeito dos métodos desenvolvidos e questões relacionadas.

Exponenciação discreta e outros problemas. Embora tenhamos nos concentrado na apresentação de métodos de transferência parcial de conhecimento a respeito dos fatores de um número, essencialmente qualquer problema em NP é adequado para a adaptação das técnicas aqui descritas. Um problema para o qual cabe destaque é o da exponenciação discreta, dada a sua fácil representação por circuitos combinacionais — e portanto, na-

tural redução ao SAT — e dada a sua vasta aplicação à Segurança da Informação. Na prática, os métodos aqui propostos podem ser facilmente adaptados para mostrar que o logaritmo discreto de um número possui um determinado prefixo, e é razoável que protocolos similares possam ser desenvolvidos para qualquer dos problemas considerados "computacionalmente difíceis" que fundamentam os atuais algoritmos criptográficos.

Furtividade da marca d'água. Uma característica que pode destacar a marca d'água dentro do artefato marcado é o fato de que ela é uma substring com propriedades bastante especiais — pelo menos no caso da marca d'água baseada no produto de primos. De fato, a substring embarcada possuirá a propriedade bastante particular de ser o produto de dois números primos grandes, enquanto a maioria dos números binários possui fatores pequenos. Isso possibilita a construção de um algoritmo simples para determinar um pequeno conjunto de substrings candidatas a marca d'água: basta procurar por substrings que representem binários sem fatores pequenos, onde a definição exata de "pequenos" poderá variar de acordo com a disposição do atacante em executar algoritmos de fatoração.

Uma saída para contornar a fraqueza acima é lançar mão de marcas d'água de tamanhos variáveis, de tal maneira que o atacante não saiba o tamanho exato da substring que deverá procurar — apenas um limite inferior. Uma outra possível saída é o uso de exponenciação discreta como a "função one-way" nos protocolos propostos.

Plágio. Marca d'água não protege contra plágio. Se um escritor mal-intencionado, após ler um livro, resolver rescrevê-lo inteiramente com suas próprias palavras, tratar-se-á, a princípio, de uma nova obra. A idéia da marca d'água é que, caso este escritor reescreva apenas trechos do livro, provavelmente, a marca d'água ficará intacta. Ou seja, o esforço para remover a marca d'água será tão grande quanto o de plagiar toda a obra. O mesmo vale para um programa de computador e um atacante que entenda toda a sua lógica e rescreva inteiramente seu código.

Hipótese Forte da Dificuldade da Fatoração. Observe que o método descrito no presente trabalho baseia-se, na verdade, numa versão modificada — de fato, ligeiramente fortalecida — da Hipótese da Dificuldade da Fatoração (HDF). A clássica HDF pressupõe que a multiplicação de números primos aleatórios é uma função one-way. Por outro lado, no protocolo proposto, um dos números primos não é perfeitamente aleatório. De fato, os bits mais significativos de um dos números primos foram obtidos deterministicamente. Assumimos que esta versão modificada do produto de primos — à qual denominamos Hipótese Forte da Dificuldade da Fatoração — também é uma função one-way. Ainda que seja perfeitamente plausível assumirmos a HFDF (e mesmo, possivelmente, prová-la), seria interessante dispor de um protocolo que se baseasse tão somente em premissas clássicas, como é o caso da HDF.

Referências

Adelsbach, A., Katzenbeisser, S., and Sadeghi, A.-R. (2003). Watermark detection with zero-knowledge disclosure. *Multimedia Syst.*, 9(3):266–278.

Adelsbach, A. and Sadeghi, A.-R. (2001). Zero-knowledge watermark detection and proof of ownership. In Moskowitz, I. S., editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 273–288. Springer.

Bento, L. M. S., Boccardo, D. R., Machado, R. C. S., de Sá, V. G. P., and Szwarcfiter, J. L. (2013). Towards a provably resilient scheme for graph-based watermarking. In Brandstädt, A., Jansen, K., and Reischuk, R., editors, *WG*, volume 8165 of *Lecture Notes in Computer Science*, pages 50–63. Springer.

Collberg, C. and Nagra, J. (2009). *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Addison-Wesley Professional, 1st edition.

Craver, S. (1999). Zero knowledge watermark detection. In Pfitzmann, A., editor, *Information Hiding*, volume 1768 of *Lecture Notes in Computer Science*, pages 101–116. Springer.

Goldwasser, S., Micali, S., and Rackoff, C. (1985). The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA. ACM.

Hamilton, J. and Danicic, S. (2011). A survey of static software watermarking. *Proc. World Congress on Internet Security*, pages 100–107.

Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*. Plenum Press, New York.

Kilian, J. (1990). *Uses of randomness in algorithms and protocols.* MIT Press.

Rabin, M. O. (1981). How to exchange secrets with oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory.

Schaefer, T. J. (1978). The complexity of satisfiability problems. In *10th annual ACM symposium on Theory of Computing*, pages 216–226. ACM.

Tseitin, G. (1983). On the complexity of derivation in propositional calculus. In Siekmann, J. and Wrightson, G., editors, *Automation of Reasoning*, Symbolic Computation, pages 466–483. Springer Berlin Heidelberg.

Venkatachalam, B. (2005). Software watermarking as a proof of identity: A study of zero knowledge proof based software watermarking. In Barni, M., Cox, I. J., Kalker, T., and Kim, H. J., editors, *IWDW*, volume 3710 of *Lecture Notes in Computer Science*, pages 299–312. Springer.

Zhu, W., Thomborson, C. D., and Wang, F.-Y. (2005). A survey of software watermarking. In Kantor, P. B., Muresan, G., Roberts, F. S., Zeng, D. D., Wang, F.-Y., Chen, H., and Merkle, R. C., editors, *Proc. IEEE Int'l Conference on Intelligence and Security Informatics*, volume 3495 of *ISI'05*, pages 454–458. Springer.

A randomized graph-based scheme for software watermarking*

Lucila Maria Souza Bento[1,2],
Davidson Rodrigo Boccardo[2],
Raphael Carlos Santos Machado[2],
Vinícius Gusmão Pereira de Sá[1],
Jayme Luiz Szwarcfiter[1,2]

[1]Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, RJ – Brasil

[2]Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO)
Duque de Caxias, RJ – Brasil

lucilabento@ppgi.ufrj.br, {drboccardo,rcmachado}@inmetro.gov.br,
vigusmao@dcc.ufrj.br, jayme@nce.ufrj.br

Abstract. *The insertion of watermarks into proprietary objects is a well-known means of discouraging piracy. It works by embedding into the object some (often surreptitious) data meant to disclose the authorship/ownership of the object. Some promising graph-based watermarking schemes to protect the intellectual property of software have been suggested in the literature, and recent efforts have been endeavored to improve their resilience to attacks. Among the pursued attributes of software watermarking solutions is the one referred to as "diversity", which is the ability to encode the intended information in many distinct forms, making it harder for an attacker to find and remove it. We introduce a graph-based scheme which achieves a high level of diversity through randomization, while admitting an efficient, linear-time implementation nonetheless.*

Resumo. *A inserção de marcas d'água em objetos proprietários é uma conhecida maneira de se desencorajar pirataria. Funciona através da inclusão de alguma informação (em geral escondida) que permita revelar autoria ou propriedade do objeto. Alguns esquemas de marca dágua baseados em grafos para proteger a propriedade intelectual de programas de computador têm sido sugeridos na literatura, e esforços recentes têm sido devotados ao aumento de sua resiliência a ataques. Entre os atributos buscados para soluções de marca d'água de programas está a chamada "diversidade", que é a habilidade de codificar a informação desejada de várias maneiras distintas, tornando mais difícil sua localização e remoção por parte do atacante. Apresentamos um esquema baseado em grafos que consegue, através de randomização, um alto grau de diversidade, permitindo, ainda assim, uma implementação eficiente em tempo linear.*

1. Introduction

For a long time have watermarks been used to enforce authenticity, authorship or ownership of objects. The rationale is that a non-authentic object would not pos-

*Work partially supported by CAPES, CNPq, FAPERJ, Pronametro 52600.017257/2013 and Eletrobrás DR/069/2012.

sess a convincing watermark lookalike, since watermarks are (ideally) hard to be counterfeit. Moreover, a watermarked object would be seriously damaged if one attempted to delete the watermark. In the early 1990's, such ancient idea has been leveraged to the context of software protection as a means to preclude—or at least discourage—the widespread crime of software piracy. A lot of research has been done on software watermarking ever since, and several distinct techniques have been used, including opaque predicates, register allocation, abstract interpretation and dynamic paths [Qu and Potkonjak 1998, Monden and Inoue 2000, Arboit 2002, Nagra and Thomborson 2004, Cousot and Cousot 2004, Collberg et al. 2004].

Graph-based watermarking schemes consist of encoding/decoding algorithms (codecs) that translate the identification data onto (and back from) some special kind of graph. The pioneering graph-based watermark for software protection was formulated in [Davidson and Myhrvold 1996]. It then inspired the publication, in [Venkatesan et al. 2001], of the first watermarking scheme in which the watermark graph is embedded into the *control flow graph* (CFG) of the software to be protected. The CFG, which can be determined by tools for static analysis of code, represents all possible sequences of computation of the program's instructions in the form of a directed graph whose vertices are the blocks of strictly sequential code, and whose edges indicate possible precedence relations between those blocks. The embedder algorithm basically creates dummy code so that new, appropriately interlinked code blocks appear in the CFG, starting at some predefined position and describing exactly the intended watermark structure.

Whereas techniques for watermark embedding are reasonably well developed by now [Collberg and Thomborson 1999, Chroni and Nikolopoulos 2012b, Bento et al. 2013a] and not in the scope of this text , the codecs that have been proposed so far still leave much room from improvement with respect to their resilience to attacks. Two attack models demand special attention, namely *subtractive attacks* and *distortive attacks* [Collberg and Nagra 2009]. In the subtractive attack model, the attacker detects the presence of the watermark and removes it altogether. This kind of attack is basically precluded by code obfuscation and suchlike techniques. The distortive attack model is in a sense more subtle, since the attacker, not being able to detect and remove the watermark as a whole, attempts to damage its structure. It can be done basically by changing the code so that some connections between code blocks disappear (in other words, by indirectly removing edges from its CFG).

The recent, ingenious codec introduced in [Chroni and Nikolopoulos 2012a] was inspired by the work of [Collberg et al. 2003]. It encodes the desired data—which we will refer to as the *key*—as an instance of the reducible permutation graphs introduced by the latter authors. It has been shown in [Bento et al. 2013b] that, even though the watermarks proposed by Chroni and Nikolopoulos manage to withstand attacks in the form of $k \leq 2$ edge removals, there is an infinite number of watermark instances generated by their codec which get irremediably damaged by $k = 3$ edge removals. The recovery of the encoded data is therefore impossible in many cases, even for a modest number of removed edges. In [Chroni and Nikolopoulos 2012c], the authors ask whether graph-based watermarks with greater resilience to attacks—as well as better time/space efficiency—could be devised.

We propose a new codec for software watermarking. The proposed codec employs

Algorithm 1: Encoding the randomized watermark

Input: an integer key ω to be encoded
Output: a randomized watermark encoding ω

1. Let B be the binary representation of ω, and let $n = |B|$. Index the bits of B, from left to right, starting from 1.
2. The watermark $G(V, E)$ is initially isomorphic to a directed path P_{n+1} on vertex set $V = \{1, \ldots, n+1\}$, i.e., the set E initially contains *path edges* from v to $w = v + 1$, denoted $[v \rightarrow w]$, for $1 \leq v \leq n$.
3. For each vertex $v \in V \setminus \{1, n+1\}$, add into E a *back edge* from v to w, denoted $[w \leftarrow v]$, where w is chosen uniformly at random from the elements of V which satisfy:
   - w is not an inner vertex of a cycle of G, and
   - $v - w$ is an *odd* (respectively, *even*) positive integer if v is the index of a bit '1' (respectively, '0') in B.

   If no such w exists, then let v remain with its current outdegree 1, i.e., do not add a back edge leaving v.

randomization to attain a high level of *diversity*, a property whose importance has been noted by the community [Collberg and Nagra 2009], and which is closely related to the resilience of the watermarks against some forms of attack. In short, the structure of the watermarks produced by our scheme is affected by random choices that are made during the execution of the encoding algorithm, which gives rise to a number of distinct graphs encoding the same piece of information. This feature makes it less likely that a watermark can be spotted through brute force comparisons—undertaken by specialized diff tools—among different watermarked programs by the same author or proprietor. Furthermore, the number of edge removals which our watermarks are able to withstand can be customized at will. That is accomplished by means of an edge-to-bit bijection, along with a decoding procedure that is immune to error propagation, making it possible that standard bit-level error-correction techniques are employed in the decoding algorithm quite straightforwardly.

The paper is organized as follows. In Section 2, we introduce the new codec. In Section 3, we propose and analyze a possible linear-time implementation for the encoding and the decoding algorithms. In Section 4, we indicate how to incorporate bit-level error-correction techniques into the new codec. In Section 5, we make our concluding remarks.

2. Randomized watermarks

We introduce a codec for graph-based watermarking of software. The codec has the following main properties:

- the encoding algorithm proceeds in a randomized fashion, therefore the same key will almost certainly give rise to distinct watermarks at different executions of the algorithm;

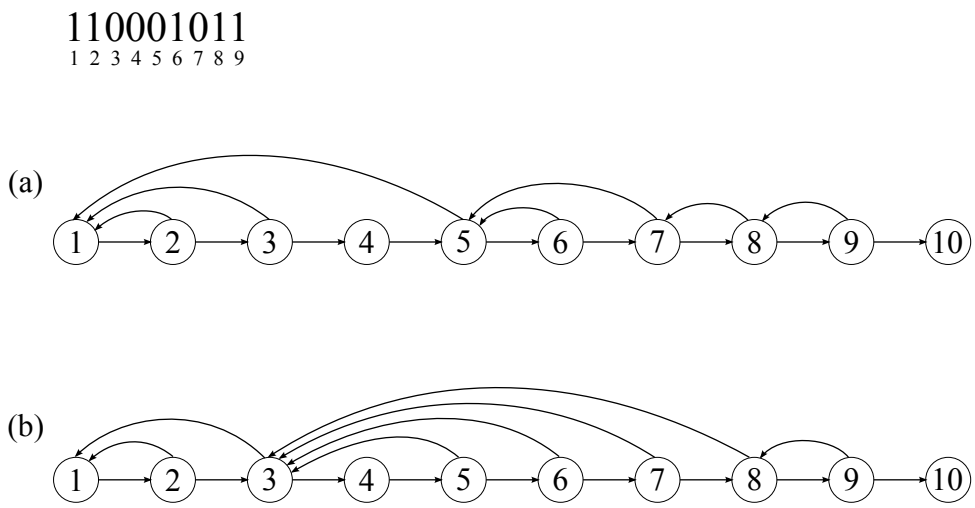


Figure 1. Distinct randomized watermarks encoding the same key $\omega = 395$

- there is a one-to-one correspondence between the edges of the watermark and the bits of the encoded key, hence distortive attacks can be detected *after* the graph-to-key decoding process, and the correction of any flipped bits—up to some predefined number— can be carried out by standard error-correction algorithms;
- both encoding and decoding algorithms can be implemented to run in linear time.

Algorithm 1 describes the basic steps of the encoding algorithm, when no extra bits—intended for error-correction—are used (we address error-correction in Section 4). If $C : v_1, v_2, \ldots, v_d, v_{d+1} = v_1$ is a cycle on d vertices of a directed graph G, we say vertices $v_2, \ldots, v_{d-1}$ are the *inner vertices* of C.

Figure 1 illustrates two watermarks generated by the new codec for key $\omega = 395$, whose binary form, determined in step 1 of the algorithm, is $B = 110001011$, with $n = 9$ bits. Both watermarks have the same number of vertices, namely $n + 1$, and both have a (unique) Hamiltonian path[1], which is created in step 2 of the algorithm. The first vertex of the Hamiltonian path, labeled 1, always corresponds to a bit '1' in B, and its outdegree is always 1. Now, each vertex from 2 to n becomes the origin of either zero or one *back edges*. The back edges with origin in each $v \in \{2, \ldots, n\}$ (or the absence thereof) will bear a one-to-one correspondence with the bits indexed from 2 to n in B: a bit '1' with index $v \geq 2$ in B gives rise to a back edge $[w \leftarrow v]$ constituting an odd-distance "backwards jump" over the Hamiltonian path (i.e., $v - w$ is odd), whereas a bit '0' either gives rise to an even-distance backwards jump or to no jumps at all (when there is no $w < v$ such that $v - w$ is even and w is not an inner vertex of a cycle).[2]

We remark that the event that a back edge cannot be added shall never occur with respect to a vertex $v \geq 2$ corresponding to a bit '1'. Indeed, because vertices are processed left-to-right by the algorithm, vertex $w = v - 1$ is never an inner vertex of a cycle by the time v is processed. Consequently, for all $v \geq 2$ corresponding to a '1', at least the back

[1] A Hamiltonian path on a graph G is a path where all vertices of G appear exactly once.

[2] All arrays in this text are 1-based, i.e. their first position has index 1.

Algorithm 2: Decoding the randomized watermark

Input: a randomized watermark G with $n+1$ vertices
Output: the key ω encoded by G

1. Label the vertices of G in ascending order as they appear in the unique Hamiltonian path of G.
2. Let B be a bit array starting with a bit '1' followed by $n-1$ bits '0'.
3. For each vertex $v \in \{2, n\}$, if there is a vertex $w < v$ such that $[w \leftarrow v] \in E(G)$ and $v - w$ is odd, then $B[v] \leftarrow$ '1'; otherwise, $B[v] \leftarrow$ '0'.
4. Return $\omega = \sum_{i=1}^{n} B[i] \cdot 2^{n-i}$.

edge $[v-1 \leftarrow v]$—an odd-distance backwards jump, as intended—will be available. On the other hand, if a vertex v corresponds to a bit '0' in B, then it is possible that no w can be the destination of a back edge with origin at v constituting an even-distance backwards jump. The absence of a back edge will therefore indicate that the bit with index v in B is a '0'. Moreover, if v gets no outgoing back edges, then vertex $v' = v+1$ is assured to get one, for if v' corresponds to a '0', then at least the edge $[v-2 \leftarrow v]$—an even-distance backwards jump, as intended—will be available, since there is no back edge closing a cycle at $v-1$.

Back to our example in Figure 1, notice that vertex 2 corresponds to a bit '1' in B and therefore receives the outgoing back edge $[1 \leftarrow 2]$, the only possible choice then. Vertex 3, on its turn, corresponds to a '0' and gets to be the origin of back edge $[1 \leftarrow 3]$, again the only possible choice. Now, vertex 4, which corresponds to a '0', must be left without an outgoing edge, for the only $w < 4$ such that $4 - w$ is even would be $w = 2$, but vertex 2 is an inner vertex of the existing cycle $1, 2, 3, 1$. Vertex 5 corresponds to a bit '0', and two back edges were available by the time it was processed, namely $[1 \leftarrow 5]$ and $[3 \leftarrow 5]$. For the watermark in Figure 1(a), the former edge was chosen, whereas the latter was chosen for the watermark in Figure 1(b). The algorithm carries on in similar fashion for vertices $6, \ldots, 9$, and the watermark is complete.

The decoding procedure consists of two steps. First, we must label the vertices of the watermark, so their correspondence to the bits of the encoded binary can be determined. This can always be done, since the blocks of the Hamiltonian path are always consecutive in the CFG, corresponding to vertices $1, 2, \ldots, n+1$. Second, we set the first bit of the binary as '1' (which is always the case, since zeroes to the left of a number are ignored), and we proceed to gathering the information encoded by the back edges, from vertex 2 onwards: a back edge $[w \leftarrow v]$ such that $v - w$ is odd (respectively, even) indicates that the bit with index v in the binary is a '1' (respectively, '0'), and vertices $2 \leq v \leq n$ which are not the origin of a back edge also correspond to bits '0' in the binary. The decoding algorithm is given in pseudocode as Algorithm 2.

3. Linear-time implementation

The two first steps of Algorithm 1 are straightforward. In order to implement step 3, however, we must be able to keep track of the current *destination candidates*, i.e. vertices $w < v$ which are not (yet) inner vertices of any cycles and therefore can (still) be picked as the destination of a back edge with origin at vertex v being currently processed. If v is even and corresponds to a bit '0', or v is odd and corresponds to a bit '1', then the destination w of the back edge $[w \leftarrow v]$ must be selected among the current even-labeled destination candidates; otherwise, w must be selected among the current odd-labeled destination candidates. Whichever the case, the algorithm must choose uniformly at random among the destination candidates whose label has the desired parity, which can be done by picking a random integer between 1 and the number of such candidates.

We employ two stacks, S_0 and S_1, each one implemented over an array so that any item can be looked up by its index in constant time. Implementing those stacks over arrays also allows for a constant-time *pop_all*(i) method, which removes all items whose indexes are greater than a given index i.[3]

The proposed implementation of step 3 consists of a loop that iterates through vertices $2, \ldots, n$ in order to determine the back edges (if any) with origin at each of these vertices, one by one. The following invariant holds: the elements in stack S_0 (respectively, S_1) are precisely the even-labeled (respectively, odd-labeled) destination candidates in ascending order (bottom-up along the stack) at any moment. All vertices $v = 1, \ldots, n$ will be added to their respective stack (even-labeled vertices into S_0, odd-labeled vertices into S_1) exactly once during the execution of the algorithm, namely by the end of the iteration during which v is visited, i.e. right after determining the destination of the back edge with origin in v.

Additionally, we need an auxiliary n-sized array, call it A, which is initially empty, and whose positions are indexed from 1 to n. Each position v of the array, for even v, will be assigned the size that stack S_1 used to have by the time v was added to stack S_0. Analogously, each position v of the array, for odd v, will be assigned the size that stack S_0 used to have by the time v was added to S_1.

Now we can describe a linear-time implementation for the whole step 3 of the encoding algorithm. Its pseudocode is depicted in Procedure 3.

After the initialization of the data structures (line 1), vertex $v = 1$ is the first to be considered. However, since no back edge with origin at vertex 1 is meant to be added, the algorithm just pushes v into S_1 (because v is odd) and writes 0 (the current size of stack S_0) to position 1 of A (line 2).[4] Now, for each vertex $v \in \{2, \ldots, n\}$, the algorithm first decides which stack—S_0 or S_1—contains the candidates among which the destination of the back edge with origin in v shall be (randomly) chosen (lines 4–7). Such stack, which will be referred to as S, will be S_0 if either an even-distance backwards jump is intended ($B[v]$ is a bit '0') and v is even, or an odd-distance backwards jump is intended ($B[v]$ is a '1') and v is odd; otherwise, the appropriate stack will be $S = S_1$. The element in

[3] That can be achieved easily by redirecting a "stack top" pointer to the stack's ith item, or, alternatively, by maintaining a "stack size" variable.

[4] This latter instruction is not actually necessary, since A was initialized with zeroes. We have included it for clarity.

Procedure 3: Determining back edges in $\mathcal{O}(n)$ time

Input: the n-bit binary representation B of the key to be encoded,
and the set E containing only the path edges of the watermark
Output: updated set E containing all (path/back) edges of the watermark

```
1.  S_0 := empty stack;  S_1 := empty stack;  A := array with n zeroes
2.  S_1.push(1);  A[1] := 0
3.  for v = 2, ..., n do
4.      if (v is even and B[v] = '0') or (v is odd and B[v] = '1') then
5.          S := S_0;  S' := S_1
6.      else
7.          S := S_1;  S' := S_0
8.      if S.size > 0 then
9.          j := integer chosen uniformly at random from [1, S.size]
10.         w := S[j]
11.         E := E ∪ {[w ← v]}
12.         S.pop_all(j)
13.         S'.pop_all(A[w])
14.     if v is even then
15.         S_0.push(v);  A[v] := S_1.size
16.     else
17.         S_1.push(v);  A[v] := S_0.size
18. return E
```

$\{S_0, S_1\} \setminus S$ will be referred to as S'. If S is empty, then there are no vertices available to be the destination of a back edge leaving v; in this case, no back edge will be added to the edge set E. Otherwise, an integer j is chosen uniformly at random between 1 and the size of S, thus determining the destination $w = S[j]$ of the back edge leaving v. Such back edge is added to E (line 11). Now, because the addition of a back edge $[w \leftarrow v]$ implies the creation of a cycle $C : w, w+1, \ldots, w+(v-w) = v, w$, all inner vertices of C which used to be destination candidates now become unavailable for future selections. In other words, they cease to be destination candidates, and must therefore be removed from their corresponding stacks, i.e. all vertices $w' > w$ must be popped from both S and S'. Since the index of w in S, call it j, is known, it is easy to remove all such vertices w' from S (line 12), for they are precisely those vertices whose indexes in S are greater than j. On the other hand, because w was never an element of S', there is no such thing as the index of w in S', and one might think that a (binary) search in S' would be called for in order to determine the index of the last element of S' that is smaller than w—which would append an extra $O(\log n)$ factor to the algorithm's time complexity. However, because w is currently *not* the inner vertex of a cycle (otherwise it would not have been selected as a back edge destination), no vertex $w' < w$ was selected as destination after w was processed, which means all vertices that belonged to S' by the time w was processed *still* belong to S', and must remain there. As a matter of fact, only those vertices must remain

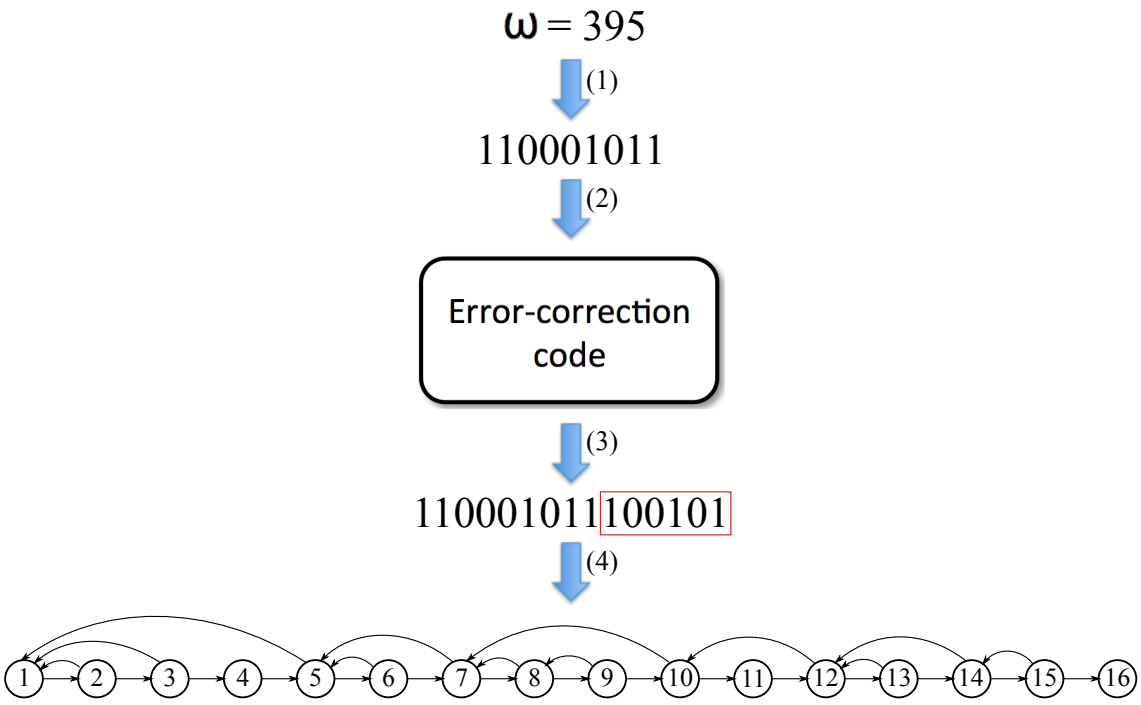


Figure 2. Encoding the key $\omega = 395$ with 1-bit error correction

in S', since vertices added to S' after w was processed are necessarily greater than w (and smaller than v), hence they have just become inner vertices of C and must be removed from S'. In other words, S' must retain precisely its r first elements, where r is the size that S' used to have when w was processed and added to S. And now the auxiliary array A comes into play, for $r = A[w]$ is precisely the value that was stored at its wth position when w was added to S (line 15 or 17, depending on the parity of w).

Since there is a constant number of operations per vertex, and all operations clearly run in $O(1)$ time, the overall complexity of Procedure 3 is $O(n)$.

4. Resilience to distortive attacks

The literature on error detection/correction techniques, mainly intended for binary messages transmitted on error-prone channels, is quite vast [Hamming 1950, Reed and Solomon 1960, Mann 1968, Purser 1995, Wicker 1995]. We do not intend to give an exhaustive account on the existing techniques in this text. Instead, we intentionally regard them as "black boxes", demonstrating how the intended results can be easily achieved. Although the existing error-correction techniques may differ (a lot) in the way they tackle a possibly damaged binary, a common requirement is the insertion of a number $f(n, t)$ of redundancy bits, for some function f. This is done in a preprocessing step of the binary representation of the key about to be encoded.

In the decoding phase of our watermarking scheme, the effect of k malicious edge removals is that of erroneously writing k or less bits '0' at positions originally occupied by bits '1' in the encoded binary. This is so because the absence of a back edge leaving vertex v, for $2 \leq v \leq n$, is regarded by the decoder as a bit '0' with index v. If a back edge leaving v used to exist in the watermark before the attack, then the bit with index v in the original binary might as well be a '1'. Yet, the consequence of each edge removal is

that of a single flipped bit (at most), because, due to the mechanics of the proposed codec, decoding errors *do not propagate*.

Suppose, on the other hand, that, instead of being based on the parity of the distance covered by backwards jumps, our encoding algorithm selected the destination w of a back edge with origin at v in the following way: pick—uniformly at random—a destination $w < v$ such that w is not an inner vertex of a cycle, and w corresponds to a bit, in the binary, that is the same as the bit in the vth position. In other words, if v is a '1', its outgoing back edge must reach a '1'; if v is a '0', its outgoing back edge must reach a '0'. Under such a codec, an edge removal resulting in the erroneous decoding of a vertex v would cascade the error to vertex v' whose outgoing back edge reached v, and to vertex v'' whose outgoing back edge reached v', and so on.

We illustrate, in Figure 2, the encoding of the same key from Figure 1, but now employing the well-known Reed-Solomon error correction technique [Reed and Solomon 1960] under a Galois field GF(2^3), which in this case provides the ability to recover from 1-bit flips (i.e., from single edge removals). In step (1), the binary form of the key is obtained; in steps (2) and (3), the binary is passed to an error-correction preprocessing step, where redundancy bits are appropriately inserted; in step (4), the final binary is translated onto the watermark graph by using the proposed encoding function (see Section 2, Algorithm 1). Notice that the preprocessing step could be made so that an arbitrary, predefined number $t > 0$ of edge flips could be afterwards detected and corrected, yet the size of the ensuing binary grows accordingly.

The decoding is done in similar fashion, as illustrated in Figure 3: in step (1), the watermark graph is decoded into the binary it represents (see Section 2, Algorithm 2); in (2) and (3), the decoded binary is passed to the error-correction post-processing step, wherefrom another binary (with unflipped bits) is produced; and, finally, in step (4), the original key is retrieved.

Thus, if the number k of missing edges is less than the fixed threshold $t > 0$ taken into consideration when preprocessing the original binary, the employed error-correction solution shall identify the flipped bits and correct them; otherwise, the attack will have succeeded in damaging the watermark permanently. In effect, no matter the chosen error-correction technique or the number t of errors the watermark can withstand, the attacker may always remove so large a number $t' > t$ of edges that no recovery is possible, as illustrated in Figure 4.

5. Conclusion

We presented a randomized codec for graph-based software watermarking. Its main property is its ability to encode the same key as distinct graphs, accounting for a high diversity, a feature whose importance has been stressed by the community. Moreover, it can be implemented to run in linear time[5] as shown in Section 3, and it is compatible with standard bit-level error-correction techniques.

For future work, it should be interesting to devise a codec with all these nice attributes, but also with the property of being embeddable in the CFG without the need

[5]An implementation using the Python language is available at `https://www.dropbox.com/s/kydbc60mkl7lf7z/randomized-watermark.py`.

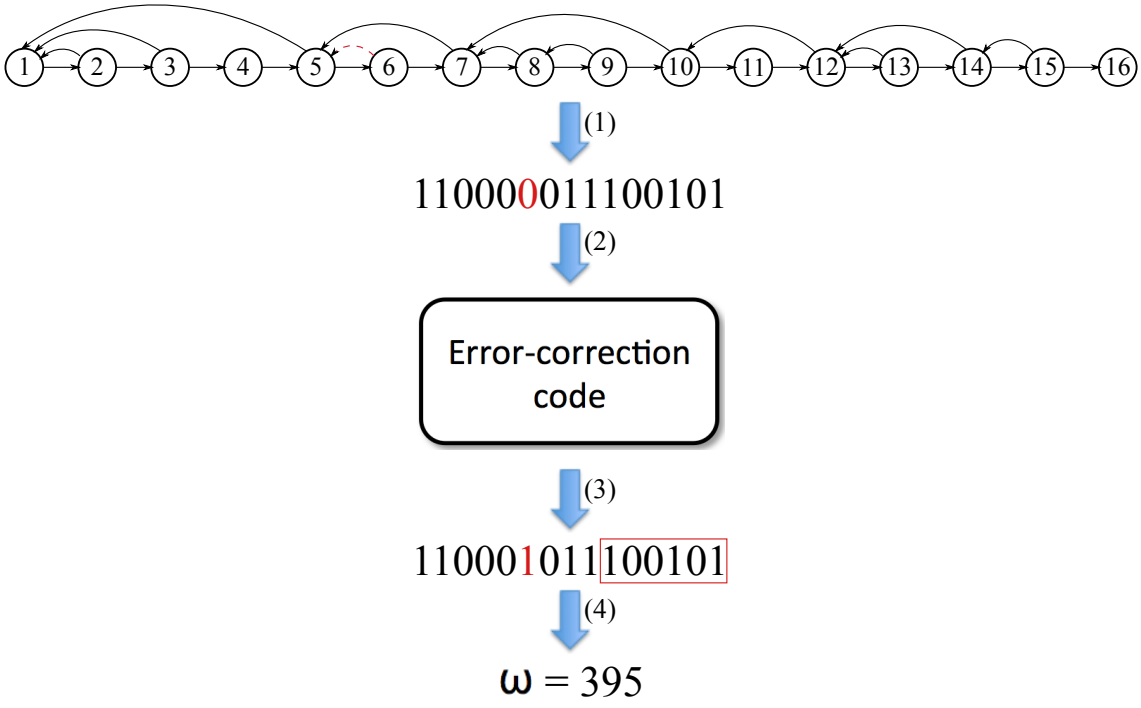


Figure 3. Decoding the watermark after the removal of edge $[5 \leftarrow 6]$

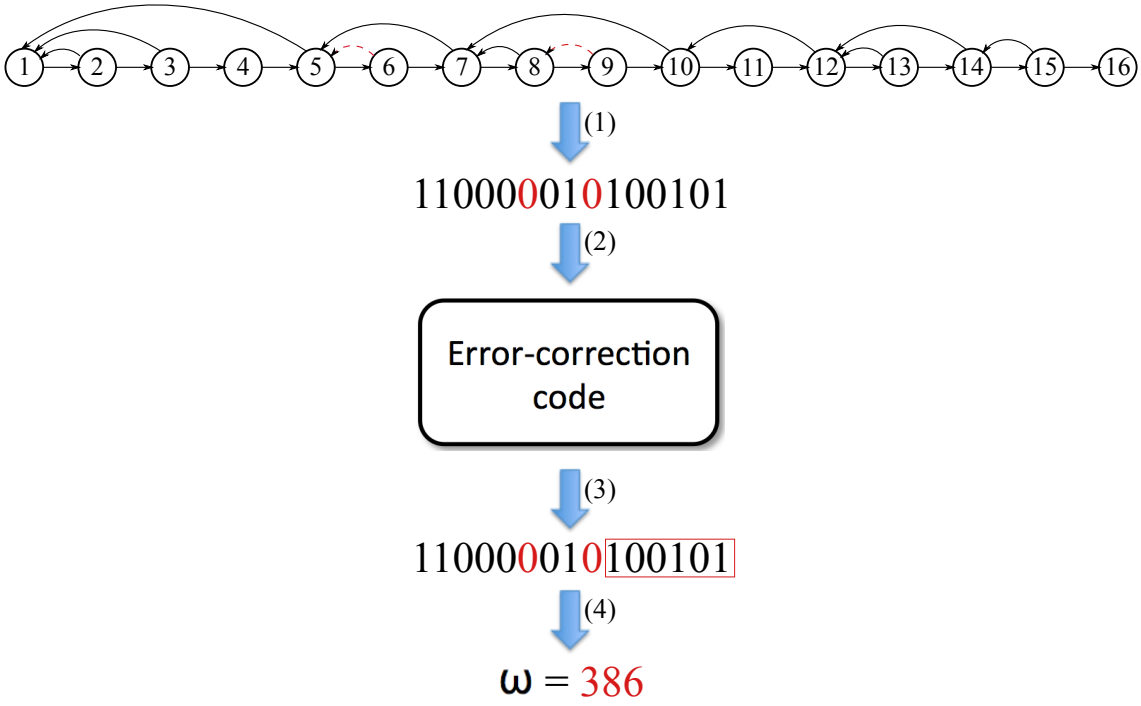


Figure 4. Decoding the watermark after the removal of edges $[5 \leftarrow 6]$ **and** $[8 \leftarrow 9]$**: an incorrect outcome is produced**

to use raw jumps (*goto* statements), something that is required not only by our codec but by all other CFG-embedding codecs we are aware of. In other words, a codec which produces watermarks that can be made to appear as subgraphs of the CFG by means of adding dummy *structured code* only. Such property would improve upon the stealthiness of the produced watermarks, since they would resemble normal, actual code even further.

References

Arboit, G. (2002). A method for watermarking Java programs via opaque predicates. In *Proc. Int. Conf. Electronic Commerce Research (ICECR-5)*.

Bento, L. M. d. S., Boccardo, D., Costa, R., Machado, R. M. S., Pereira de Sá, V. G., and Szwarcfiter, J. L. (2013a). Fingerprinting de software e aplicações à metrologia legal. In *Proc. 10th International Congress on Electrical Metrology (SEMETRO'13)*.

Bento, L. M. S., Boccardo, D. R., Machado, R. C. S., Pereira de Sá, V. G. a. P., and Szwarcfiter, J. L. (2013b). Towards a provably resilient scheme for graph-based watermarking. In *Proc. Workshop on Graph-Theoretic Concepts in Computer Science (WG'13), LNCS 8165*, pages 50–63. Springer.

Chroni, M. and Nikolopoulos, S. D. (2012a). An efficient graph codec system for software watermarking. In *36th IEEE Conference on Computers, Software, and Applications (COMPSAC'12)*, pages 595–600. IEEE Proceedings, 36th edition.

Chroni, M. and Nikolopoulos, S. D. (2012b). An embedding graph-based model for software watermarking. In *Proc. International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'12)*, pages 261–264. IEEE.

Chroni, M. and Nikolopoulos, S. D. (2012c). Multiple encoding of a watermark number into reducible permutation graphs using cotrees. In *CompSysTech*, pages 118–125.

Collberg, C., Carter, E., Debray, S., Huntwork, A., Linn, C., and Stepp, M. (2004). Dynamic path-based software watermarking. In *Proc. Conference on Programming Language Design and Implementation (SIGPLAN'04)*.

Collberg, C., Kobourov, S., Carter, E., and Thomborson, C. (2003). Error-correcting graphs for software watermarking. In *Proc. 29th Workshop on Graph-Theoretic Concepts in Computer Science (WG'03), LNCS 2880*, pages 156–167. Springer.

Collberg, C. and Nagra, J. (2009). *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Addison-Wesley Professional.

Collberg, C. and Thomborson, C. (1999). Software watermarking: Models and dynamic embeddings. In *Proc. 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL'99, pages 311–324. ACM.

Cousot, P. and Cousot, R. (2004). An abstract interpretation-based framework for software watermarking. In *Proc. Conference Record of the 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 173–185. ACM Press, New York, NY.

Davidson, R. and Myhrvold, N. (1996). Method and system for generating and auditing a signature for a computer program. US Patent 5,559,884.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160.

Mann, H. (1968). *Error Correcting Codes*. John Wiley and Sons.

Monden, A. and Inoue, K. (2000). A practical method for watermarking Java programs. In *Proc. 24th Computer Software and Applications Conference*, pages 191–197.

Nagra, J. and Thomborson, C. (2004). Threading software watermarks. In *Proc. 6th International Workshop on Information Hiding, LNCS 3200*, pages 208–233. Springer.

Purser, M. (1995). *Introduction to Error-Correcting Codes*. Artech House Inc.

Qu, G. and Potkonjak, M. (1998). Analysis of watermarking techniques for graph coloring problem. In *ICCAD*, pages 190–193.

Reed, I. S. and Solomon, G. (1960). Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304.

Venkatesan, R., Vazirani, V. V., and Sinha, S. (2001). A graph theoretic approach to software watermarking. In *Proc. 4th International Workshop on Information Hiding (IHW'01)*, pages 157–168. Springer.

Wicker, S. B. (1995). *Error control systems for digital communication and storage*. Prentice Hall.

Esquema de Estruturação SbC-EC para Log Seguro

Sérgio Câmara[1,2], **Luci Pirmez**[1], **Luiz F.R.C. Carmo**[1,2]

[1]Programa de Pós-Graduação em Informática - Instituto Tércio Pacitti / IM
Universidade Federal do Rio de Janeiro, 21.941-901, RJ – Brasil

[2]Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro)
Av. Nossa Senhora das Graças, 50, Xerém, Duque de Caxias, 25250-020, RJ – Brasil

{smcamara,lfrust}@inmetro.gov.br, luci@nce.ufrj.br

Abstract. *Secure log schemes ensure detection of possible attacks against audit logs located in devices under an unprotected environment. This paper describes the structure scheme SbC-EC for secure logging, suitable for storage and network communication constrained devices, presenting two new features: Split by Category and Entry Compaction. We also describe the SbC-EC MAC secure log scheme, which implements the new proposed structure along with symmetric cryptography primitives and the FssAgg authentication scheme for protecting the log files. SbC-EC MAC presents a storage gain in comparison to other existing symmetric secure log schemes.*

Resumo. *Esquemas de log seguro garantem a detecção de possíveis ataques contra o log de auditoria residente em dispositivos de um ambiente não protegido. Este artigo descreve o esquema de estruturação SbC-EC para log seguro, apropriado para dispositivos com restrições de armazenamento e de comunicação em rede, apresentando duas novas características: Separação por Categoria e Compactação de Entradas. Descrevemos também o esquema de log seguro SbC-EC MAC, que implementa a nova estruturação proposta em conjunto com primitivas da criptografia simétrica e o esquema de autenticação FssAgg para a proteção dos arquivos de log. O SbC-EC MAC apresenta um ganho de armazenamento em relação a outros esquemas simétricos de log seguro.*

1. Introdução

O log de auditoria é um importante mecanismo da computação forense, capaz de oferecer segurança e confiabilidade aos sistemas computacionais. No log de auditoria podem ser registrados, por exemplo, eventos críticos como o estado e erros de execução de programas, atividades de rede e de usuários e modificação de dados. Dessa forma, um arquivo de log pode ser utilizado para efetuar a análise e diagnóstico de incidentes de segurança, permitindo a rastreabilidade da causa raiz de um problema, ou ataque, no sistema.

Devido aos benefícios obtidos com a implementação do log de auditoria e por ser de grande valia para uma análise forense, os arquivos de log são constantes alvos de ataques após a invasão de um sistema. Um atacante experiente espera apagar, ou modificar, qualquer informação registrada capaz de revelar sua identidade ou permitir a detecção de traços do comprometimento do sistema. Além disso, o atacante espera

*Artigo financiado com recursos do projeto Pronametro 52600.011757/2014.

manter os métodos de ataque transparentes para o administrador do sistema, escondendo as brechas de segurança para serem utilizadas em um futuro ataque [Bellare e Yee 1997]. Dessa forma, torna-se essencial manter o log de auditoria íntegro e seguro contra qualquer tipo de manipulação maliciosa. Uma solução possível para garantir a proteção ao log de auditoria inclui o uso de técnicas de hardware que são resistentes à violação (*tamper-proof*). Entretanto, os componentes de hardware demandados por essas técnicas podem onerar muito o custo de produção de um dispositivo com restrições de recursos como memória e processamento (p. ex., sensores sem-fio) [Ma e Tsudik 2007]. Outra técnica alternativa é a adoção de armazenamentos controlados por software do tipo WORM, *Write Once Read Multiple*, que garantem que uma informação uma vez escrita não pode ser modificada posteriormente. No entanto, esses armazenamentos são vulneráveis às ações maliciosas de atacantes internos com alto grau de privilégio e acesso físico aos discos [Oprea e Bowers 2009].

Outra técnica considerada para garantir a proteção ao log de auditoria é a utilização de um servidor *online* de coleta, responsável por requisitar e armazenar em tempo real as entradas de log dos dispositivos. Essa abordagem possui algumas desvantagens como a dependência de uma comunicação sempre ativa entre os dispositivos e o servidor central de coleta, o que pode não ser factível em determinados tipos de cenários. Com isso, a comunicação incerta, ou não frequente, com o servidor de coleta, além do armazenamento local por tempo indeterminado das entradas de log nos dispositivos, são condições que precisam ser consideradas. Ao longo do tempo, alguns protocolos e técnicas criptográficas foram criados e aprimorados a fim de garantir a proteção dos arquivos de log nos dispositivos, desenvolvendo, portanto, os mecanismos de logs seguros. Os logs seguros são mecanismos de segurança que possuem propriedades como *forward-security*[1] , ser não falsificável, ser a prova de violação, oferecer verificação de integridade (pública ou não), entre outras adicionais como prover confidencialidade e controle de acesso aos registros, verificação individual de entradas e busca por palavras-chave.

Juntamente com a necessidade de segurança, os arquivos de log dos dispositivos tendem a crescer indefinidamente com o passar do tempo, podendo conter milhares de entradas de logs, necessitando cada vez mais espaço de armazenamento [Ma e Tsudik 2009b]. Além disso, as boas práticas do padrão *ISA - Security for industrial automation and control systems*, antiga ISA-99, determinam, como requisito obrigatório, que sistemas embarcados devem oferecer uma capacidade de armazenamento suficiente para registrar os eventos auditáveis, a fim de que não haja perda de informações entre uma auditoria e outra [International Electrotechnical Commission 2011]. Logo, em um cenário onde os dispositivos apresentam restrições de memória e de comunicação em rede, as condições de um arquivo de log sempre crescente e o requisito de um armazenamento suficiente conflitam diretamente.

Com base nos problemas de segurança e armazenamento de arquivos de log citados, o presente trabalho descreve um novo esquema de estruturação de log seguro apropriado para dispositivos com restrições de memória de armazenamento e de comunicação em rede (p. ex., em aplicações de sistemas ciber-físicos, sistemas embarcados, etc). O esquema de estruturação SbC-EC apresenta duas características com o objetivo de reduzir

[1]Propriedade que assegura o não comprometimento do uso de chaves utilizadas no passado a partir da revelação de qualquer informação no momento presente [Bellare e Yee 2003].

o espaço necessário para armazenar entradas de log: (i) Separação por Categoria (*Split by Category*, SbC) e (ii) Compactação de Entradas de log (*Entry Compaction*, EC). Além disso, apresentamos um novo esquema de log seguro que utiliza essa nova estruturação, o esquema SbC-EC MAC. O SbC-EC MAC utiliza o esquema de autenticação FssAgg em conjunto com outros mecanismos para estabelecer a segurança necessária ao log de auditoria. Além disso, avaliamos o SbC-EC MAC juntamente aos esquemas de Schneier-Kelsey [Schneier e Kelsey 1998] e FssAgg MAC [Ma e Tsudik 2007], a fim de mostrar que ele permite um maior ganho de espaço de armazenamento em relação a esses dois.

O restante do artigo está organizado da seguinte maneira. A seção 2 destaca os trabalhos importantes na área de logs seguros, mencionando as principais características dos esquemas e possíveis desvantagens. A seção 3 descreve o modelo geral do sistema adotado no trabalho e o modelo de atacante, detalhando seus modos de operação. Na seção 4 é descrita a proposta do novo esquema de estruturação SbC-EC para log seguro e suas novas características. Em seguida, apresentamos um novo esquema de log seguro, o SbC-EC MAC. Cada uma de suas funções e os passos detalhados de funcionamento são descritos na seção 5. A seção 6 apresenta uma discussão sobre a segurança do SbC-EC MAC, incluindo suas propriedades e possíveis vulnerabilidades. Na seção 7, realizamos uma avaliação comparativa entre o nosso esquema de log e outros dois esquemas de log seguro da literatura e, por fim, concluímos nosso trabalho e indicamos nossos esforços futuros na seção 8.

2. Trabalhos Relacionados

A seguir, destacamos os principais trabalhos relacionados à elaboração de esquemas e características de logs seguros. Diferentemente da abordagem na qual os dispositivos possuem comunicação com um servidor *online* 100% do tempo, estes trabalhos tem como foco os esquemas para proteção dos logs "em repouso" (*logs at rest*). Esses esquemas oferecem soluções tanto no domínio da criptografia simétrica quanto na de chave-pública. Os trabalhos de Bellare e Yee foram os primeiros a incorporar as características de *forward security* aos logs de auditoria a fim de garantir posteriormente a integridade do fluxo das entradas de log (*forward-secure stream integrity*). Eles montam seu esquema utilizando MACs (*Message authentication codes*), onde as entradas de log são indexadas de acordo com períodos de tempo [Bellare e Yee 1997] [Bellare e Yee 2003].

O esquema Schneier-Kelsey descreve um esquema de chave simétrica utilizando cadeia de hash autenticada por MACs para estabelecer uma dependência entre as entradas de log, permitindo detectar qualquer alteração feita na ordem das entradas. O trabalho também descreve uma implementação do protocolo para sistemas distribuídos, incluindo troca de mensagens, criação e fechamento de arquivo de log [Schneier e Kelsey 1998] [Schneier e Kelsey 1999]. Uma das desvantagens do esquema Schneier-Kelsey é a sobrecarga na comunicação e no armazenamento das entradas de log, uma vez que para cada entrada é guardado um MAC para autenticação individual. Além disso, o esquema é suscetível ao ataque de *truncation*, no qual o atacante é capaz de eliminar *n* últimas entradas do log sem que essa manobra seja percebida em uma verificação posterior.

O Logcrypt é uma extensão do esquema de Schneier-Kelsey utilizando criptografia de chave-pública [Holt 2006]. Da mesma forma, esse esquema apresenta uma grande sobrecarga na comunicação, por estabelecer frequentes trocas de pares de cha-

ves entre os dispositivos e o servidor central, e no armazenamento, em consequência das assinaturas digitais para uma, ou mais, entradas de log. O Logcrypt também é vulnerável ao ataque de *truncation*. O trabalho de Accorsi descreve o BBox, uma caixa-preta para armazenamento do log de auditoria em sistemas distribuídos com apoio de componentes de computação confiável, como o TPM (*Trusted Plataform Module*). O esquema de log seguro que compõe o BBox segue a mesma linha do esquema de Schneier-Kelsey, utilizando cadeias de hash assinadas digitalmente [Accorsi 2011] [Accorsi 2013]. Entretanto, essa abordagem também oferece desvantagens em relação à sobrecarga de comunicação/armazenamento devido às assinaturas de cada entrada de log. Além disso, o modelo centralizador adotado não oferece segurança apropriada aos logs nos dispositivos periféricos, os quais podem ser comprometidos de maneiras não previstas no estudo.

Ma e Tsudik desenvolveram o primeiro esquema de autenticação de log seguro apropriado para dispositivos com restrições de comunicação e armazenamento. O esquema de autenticação *FssAgg* (*Forward-Secure Sequential Aggregate*), proposto tanto para a criptografia convencional quanto para a de chave-pública, permite que o dispositivo assinante combine diferentes tags de autenticação criadas em diferentes chaves/períodos de tempo em uma única tag de tamanho constante. Dessa forma, a autenticação do arquivo de log depende de apenas uma tag, resistente ao ataque de *truncation* devido à propriedade "all-or-nothing" de verificação de assinatura [Ma e Tsudik 2007] [Ma e Tsudik 2008] [Ma e Tsudik 2009b]. O esquema SbC-EC MAC, apresentado no presente trabalho, utiliza a autenticação *FssAgg* a fim de compor a tag única para o verificador.

Yavuz et al. desenvolveram o esquema BAF (*Blind Aggregation Forward*), aplicando também a autenticação *FssAgg*. O BAF oferece maneiras menos custosas de realizar a assinatura digital das entradas de log, sendo, portanto, uma opção para dispositivos com baixo poder computacional [Yavuz e Ning 2009] [Yavuz et al. 2012]. Em particular, o nosso trabalho descreve esquemas para sistemas com restrições de em espaço de armazenamento e banda de rede. Dessa forma, optamos por um esquema de autenticação baseado em criptografia simétrica por se mostrar mais adequada aos nossos propósitos.

O presente trabalho se diferencia dos demais na forma de como estruturar o log seguro usado pelos dispositivos. Enquanto os outros trabalhos organizam as entradas de log de acordo com o tempo em que foram geradas e em uma única sequência, nossa estruturação propõe uma organização pelo tempo da geração das entradas porém separadas em categorias de tipos de log. Além disso, essa divisão permite que a verificação do log de auditoria seja feita em subconjuntos de entradas de log, ou seja, por uma categoria de cada vez. Isso oferece, portanto, uma opção intermediária de verificação entre a verificação integral do arquivo de log, quando não necessário, e a verificação individual de entradas (p. ex., *immutability* [Ma e Tsudik 2009b]), onde a última introduz uma sobrecarga de armazenamento para cada uma das entradas.

3. Modelos

Nesta seção, é apresentada a descrição do Modelo do Sistema (seção 3.1) e do Atacante (seção 3.2) utilizados neste trabalho.

3.1. Modelo do Sistema

O modelo de sistema utilizado nesse trabalho é baseado no modelo descrito por Schneier [Schneier e Kelsey 1999]. O sistema é composto basicamente por três partes, T (máquina

confiável), U (máquina não confiável) e V (verificador semi-confiável), que são detalhadas em seguida:

1. T, a máquina confiável. T pode ser instanciado de diversas maneiras, p. ex., como um servidor em um ambiente controlado. T possui armazenamento suficiente para guardar todos os logs de U e autoriza um verificador V a ter acesso aos logs de U. Por fim, T realiza a verificação final dos arquivos de log, uma vez transmitidas a ele.
2. U, a máquina não confiável. Não há garantias de que essa máquina está totalmente segura contra atacantes, erros de software, entre outras ameaças. Nela são gerados e armazenados os logs por um período de tempo indeterminado. Uma vez comprometido, U agirá da forma que o atacante estipular. U é capaz de se comunicar e interagir com T e V.
3. V, o verificador semi-confiável. O verificador pode realizar a função de inspecionar a integridade do arquivo de log de auditoria, porém sem alterar nenhuma entrada. Em geral, V se caracteriza por um pequeno grupo de entidades autorizadas (p. ex., administradores de sistemas) a obter e ler uma cópia dos logs de U, quando necessário.

Durante o período de inicialização do sistema é realizada uma comunicação inicial entre o servidor T e o dispositivo U a fim de compartilhar os segredos iniciais do esquema de log seguro. Assumimos que todas as transmissões entre as partes do sistema sejam estabelecidas por uma comunicação segura, utilizando protocolos reconhecidos de autenticação e de troca-de-chaves.

O dispositivo U gera entradas de log de formato bem definido e as guarda em seu armazenamento não volátil. Em intervalos de tempo não frequentes e não determinísticos, o dispositivo U transmite seus logs armazenados para o servidor T, que, por sua vez, verifica a integridade das informações enviadas e as guarda de forma segura. Além da capacidade de U de armazenar logs, ele também é capaz de apagar informações da memória a fim de cumprir os protocolos de autenticação. Assumimos, também, que U é capaz de gerar números pseudo-aleatórios de maneira robusta.

Abaixo apresentamos as primitivas criptográficas e suas notações usadas ao longo do artigo:

1. $prf(K_0)$ é uma *pseudo-random function*, capaz de transformar a chave K_0 para K_1.
2. $\mathcal{H}(X)$ é uma *one-way function* que recebe uma mensagem X e calcula um resumo de tamanho fixo. $\mathcal{H}$ pode ser implementado com SHA-1 [National Institute of Standards and Technology 2012].
3. $E_{K_0}(X)$ é a cifragem simétrica de X usando a chave K_0. E pode ser um algoritmo como o AES [National Institute of Standards and Technology 2001].
4. $MAC_{K_0}(X)$ é o *message authentication code* simétrico de X usando a chave K_0. Pode ser implementado com HMAC [Bellare et al. 1996].

3.2. Modelo de Atacante

Como dito anteriormente, os logs de auditoria são alvos importantes para atacantes que queiram esconder traços de seu ataque, agindo de forma a apagar, modificar, inserir ou reordenar as entradas de log a fim de atingir seu objetivo. De fato, os esquemas de log

seguro não protegem contra ações maliciosas no arquivo de log. A característica principal de um esquema de log seguro é garantir que qualquer modificação indevida realizada seja detectada no ato da verificação do arquivo de log.

A seguir, consideramos as possibilidades de um atacante durante o comprometimento de U. Dada a última transmissão de entradas de log de U para T no instante de tempo t_1 e que o comprometimento de U acontece no instante de tempo t_2, é assumido que:

- O atacante toma conhecimento das chaves A_{t_2} e B_{t_2}, guardadas em U, no momento do comprometimento e é capaz de derivar todas as evoluções futuras dessas chaves.
- O atacante é capaz de gerar as tags do verificador e da parte confiável, uma vez que o algoritmo de geração de tags é público.
- As entradas de log que foram geradas em U, após o instante de tempo t_1, não podem ser consideradas confiáveis. O atacante tem a capacidade de apagá-las, modificá-las, reordená-las ou inserir novas entradas.
- O atacante desconhece as chaves geradoras das entradas de log $L_{t_1}...L_{t_2-1}$. Portanto, qualquer manipulação realizada nessas entradas poderá ser detectada em uma futura verificação do log por V ou T.
- Qualquer entrada de log gerada após L_{t_2}, inclusive, poderá ser manipulada a fim de contornar uma detecção posterior.

4. Esquema de Estruturação SbC-EC para Log Seguro

Um dos desafios relacionados à geração de logs de auditoria em dispositivos em geral é a necessidade de armazenamento de um grande, e crescente, volume de informação ao longo do tempo de operação. Com o objetivo de diminuir o espaço necessário para o armazenamento dessas informações, nesse trabalho é proposto um novo esquema de Estruturação para log seguro, denominado SbC-EC, que apresenta uma nova maneira de organizar semanticamente as entradas de log.

A estratégia principal adotada no esquema de estruturação SbC-EC reside na eliminação de informação redundante encontrada nos *payloads* das entradas de log semelhantes. Uma vez que o mecanismo de log seguro detém a propriedade de *forward-integrity*, é impossível realizar alterações nas entradas passadas de forma manter a integridade do fluxo de entradas, logo, qualquer alteração de *payload* deve ser executada na entrada antes de ser logada. Para eliminar informações redundantes do *payload*, a entrada a ser logada L_i deverá referenciar alguma anterior semelhante a ela, porém, isso torna-se impeditivo quando o número de entradas é muito grande, necessitando realizar buscas cada vez maiores para logar uma só entrada. Dessa forma, limitamos a comparação apenas a um nível, ou seja, caso o *payload* de L_i for semelhante ao de L_{i-1}, L_i é logada de forma compactada. Essa manobra estabelece a propriedade de *Compactação de Entradas* (*Entry Compaction*, EC) nesse esquema de estruturação.

A *Compactação de Entradas* funciona da seguinte maneira. Consideramos um *payload* bem estruturado P_n, composto pela tupla $(ID_n, M_n, [Params_n], TS_n)$, onde ID_n é o identificador único da mensagem de log, M_n é a mensagem de log, $[Params_n]$ é o conjunto de parâmetros referente à mensagem de log, caso haja, e TS_n é o *timestamp* da geração do *payload*. Quando um novo payload P_i está para ser logado, é verificado se $ID_i = ID_{i-1}$, caso seja verdade, o *payload* P_i será composto pela tupla

$(TS_{dif}, [Params_i])$, onde TS_{dif} é a diferença horária entre o *timestamp* de P_i e o *timestamp* de P_{i-1} ($TS_{dif} = TS_i - TS_{i-1}$).

Uma vez que os eventos auditáveis são de diversos tipos, é baixa a probabilidade de que, em um único fluxo de entradas, duas entradas semelhantes sejam logadas consecutivamente. A partir disso, desejamos aumentar essa probabilidade e, consequentemente, compactar mais entradas e poupar mais espaço de armazenamento. Para que isso seja possível, a característica de *Separação por Categoria* (*Split by Category*, SbC) foi estabelecida. A estruturação SbC-EC classifica cada entrada de log gerada em uma das $1..c$ categorias pré-estabelecidas e a concatena junto às demais da mesma categoria, em diferentes fluxos de entradas de log. Cada fluxo corresponde a um arquivo de log.

Assumimos que um componente de software saiba identificar qual a categoria de cada *payload* de log antes de ser concatenada como uma entrada no arquivo de log. Uma categoria de log corresponde aos grupos de registros de eventos auditáveis semelhantes, tais como eventos de controle de acesso, erros de requisição, eventos de sistema, alteração de configuração, potenciais atividades de reconhecimento e eventos de log [International Electrotechnical Commission 2011].

5. Esquema de Log Seguro SbC-EC MAC

Nessa seção é apresentado um novo esquema de log seguro, o SbC-EC MAC, utilizando criptografia simétrica e com base na estruturação SbC-EC. A criptografia simétrica é aqui aplicada pois oferece uma sobrecarga de autenticação menor do que a criptografia assimétrica em relação à quantidade de dados necessários. Além disso, o SbC-EC MAC utiliza o esquema simétrico de autenticação *FssAgg* [Ma e Tsudik 2009b] adaptado à nova estruturação, a fim de gerar as tags de autenticação necessárias das entradas de log. O esquema *FssAgg* estipula uma tag única (assinatura agregada) a fim de autenticar todas as entradas de log existentes em um arquivo de log.

A seguir, são descritos os algoritmos presentes no esquema SbC-EC MAC:

- *SbC-EC.Skg – Secret key generation*, o algoritmo de geração de chave secreta é usado para gerar uma chave inicial de tamanho seguro a ser usada pelo algoritmo de criptografia simétrica.
- *SbC-EC.Ckg – Category key generation*, o algoritmo de geração das chaves iniciais de verificador para cada categoria. O algoritmo recebe como entrada uma chave inicial de verificador A_0 e o número de categorias c e responde, como saída, as chaves de verificador $A_{1,1}, A_{1,2}, ..., A_{1,c}$, uma para cada categoria.
- *SbC-EC.Vsig – Verifier signature*, o algoritmo assina-e-agrega (*sign-and-aggregate*) a fim de formar a tag de autenticação do verificador. O algoritmo recebe como entradas uma chave secreta, uma mensagem a ser assinada e a tag de autenticação computada até aquele ponto. Ele calcula a assinatura da mensagem e a agrega à tag já existente. Ao final, o algoritmo realiza uma atualização da chave secreta usada. O *SbC-EC.Kupd*, *Key update*, é realizado dentro do algoritmo de assinatura por questões de segurança. Esses algoritmos remetem aos *FssAgg.Asig* e *FssAgg.Upd* descritos por Ma e Tsudik.
- *SbC-EC.Tsig – Trusted party signature*, o algoritmo que gera a tag de autenticação da parte confiável. Ele recebe como entradas uma chave secreta e c tags de autenticação, $c \geq 1$. O algoritmo concatena e assina as tags da entrada para

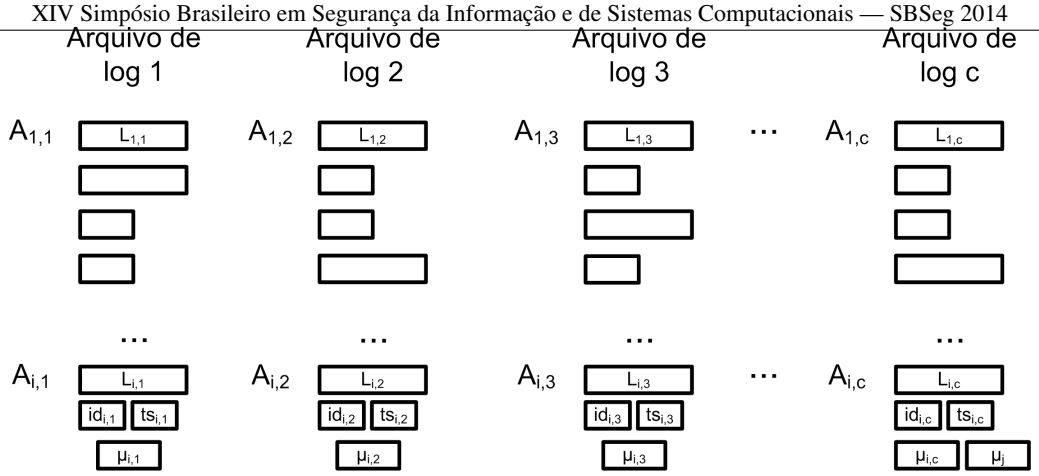


Figura 1. Representação dos elementos do SbC-EC MAC.

gerar uma única tag. Por fim, o algoritmo atualiza a chave secreta, utilizando *SbC-EC.Kupd*, por questões de segurança.

- *SbC-EC.Vver – Verifier verification*, o algoritmo de verificação da parte verificadora V recebe uma chave secreta, um fluxo de entradas de log e sua tag de autenticação e responde se a sequência de entradas está íntegra ou não. Esse algoritmo remete ao *FssAgg.Aver*.
- *SbC-EC.Tver – Trusted party verification*, o algoritmo de verificação da parte confiável recebe uma chave secreta, c fluxos de entradas de log em conjunto com suas c tags de autenticação, $c \geq 1$, e uma tag de autenticação da parte confiável. O algoritmo recalcula a tag da parte confiável a partir das entradas e responde, caso essa tag for igual à tag da parte confiável informada, os fluxos de log estão íntegros, caso contrário, não.

Um dos mecanismos usados pelo SbC-EC MAC é o de evolução de chaves secretas, usada também em outros esquemas de log seguro e em aplicações que exigem propriedades como *forward-security*. Esse mecanismo permite que uma chave K_i evolua para K_{i+1} de forma determinística através de uma função pseudo-randômica $prf(K_i)$. É impossível obter K_i a partir de K_{i+1}.

No SbC-EC MAC temos que, para cada arquivo de log representando uma categoria m, onde $1 \leq m \leq c$, existe uma tag única de verificador. Essa tag de verificador é representada por $\mu_{i,m}$, correspondente à i-ésima entrada de log da categoria m. Existe, também, uma única tag da parte confiável μ_j, correspondente à j-ésima evolução da chave secreta da parte confiável B_j. A Figura 1 mostra uma representação dos diferentes arquivos de log, cada um com suas correspondentes entradas de log, compactadas ou não, as chaves secretas de evolução e tags de verificador, e a única tag da parte confiável existente no esquema.

A seguir, descrevemos os passos do esquema SbC-EC MAC desde a preparação e inicialização dos arquivos de log, a concatenação e compactação de novas entradas de log e, por fim, as maneiras de realizar as verificações do log de auditoria para as diferentes partes do sistema.

5.1. Inicialização do log de auditoria

O dispositivo U deve efetuar alguns passos de preparação do arquivo de log antes de criá-lo. Primeiramente, U deve utilizar o algoritmo *SbC-EC.Skg* para gerar duas chaves

secretas: (i) A_0, chave inicial do verificador V e (ii) B_0, chave inicial da parte confiável T. U deve definir c, o número de categorias de logs, podendo esse dado estar predefinido e protegido no dispositivo. A partir disso, U deve informar a T esses 3 dados (A_0, B_0, c). Assumimos que a comunicação entre U e T seja segura, autenticada e à prova de ataques.

Após a confirmação de que T recebeu os dados enviados, U utiliza a função *SbC-EC.Ckg* para gerar todas as chaves iniciais de verificador de cada categoria. A partir disso, inicializa-se c arquivos de log, um para cada categoria. Para garantir a detecção de um ataque de deleção total do arquivo de log, antes de qualquer entrada legítima ser guardada, uma entrada "*dummy*" é gerada e concatenada em todos os arquivos. Essa medida evita que, após um atacante comprometer U e apagar todas as entradas de log existentes até o momento, ele não consiga esconder esse fato reivindicando que o arquivo de log nunca foi inicializado.

5.2. Concatenação e compactação de entradas de log

Para cada última entrada de cada arquivo de log, são guardados em claro (ou seja, sem estar criptografados) os dados $ID_{i-1,m}$ e $TS_{i-1,m}$, isso se justifica pelo processo de compactação de entradas, como explicado na seção 4. Descrevemos, a seguir, os procedimentos necessários durante o procedimento de concatenação de uma nova entrada de log. Primeiramente, é descrita a concatenação de uma entrada normal (quando $ID_i \neq ID_{i-1}$) e, em seguida, a concatenação de uma entrada compactada (quando $ID_i = ID_{i-1}$).

A partir da geração de um novo *payload* P, uma função decisora de categoria será responsável por escolher qual arquivo de log m deve concatenar a nova entrada. Uma vez definida a categoria, inicia-se o procedimento de concatenação da entrada. Assumimos que o arquivo de log da categoria m já possui as entradas $L_{1,m}$, $L_{2,m}$, $L_{3,m}$, ..., $L_{i-1,m}$. Os passos necessários para concatenar a entrada $L_{i,m}$ não compactada são descritos a seguir (Figura 2):

1. ID_i é comparado com ID_{i-1}. Como são diferentes nesse caso, ID_i é guardado e ID_{i-1} é apagado.
2. TS_i é guardado e TS_{i-1} é apagado.
3. $L_{i,m} = E_{A_{i,m}}(P_i)$, gera a nova entrada a partir da cifragem simétrica do *payload* P_i com a chave $A_{i,m}$.
4. Gera a i-ésima assinatura agregada $\mu_{i,m}$, tag do verificador, calculada conforme a Equação 1 [Ma e Tsudik 2009b]:

$$\mu_{i,m} = \mathcal{H}(\mathcal{H}(...\mathcal{H}(\mu_{1,m}||MAC_{A_{1,m}}(L_{1,m}))...)||MAC_{A_{i,m}}(L_{i,m})) \quad (1)$$

5. *SbC-EC.Kupd($A_{i,m}$)*, evolui a chave $A_{i,m}$ para $A_{i+1,m}$.
6. Gera a j-ésima tag da parte confiável, calculada conforme a Equação 2:

$$\mu_j = MAC_{B_j}(\mu_1||\mu_2||...||\mu_c) \quad (2)$$

   onde μ_m é a atual tag de verificador da categoria m, $1 \leq m \leq c$.
7. *SbC-EC.Kupd(B_j)*, evolui a chave B_j para B_{j+1}.

Para a concatenação de uma nova entrada de log $L_{i,m}$ em que ID_i é igual ao da última entrada $L_{i-1,m}$, a nova entrada será compactada. Os passos necessários para concatenar $L_{i,m}$ compactada são descritos abaixo:

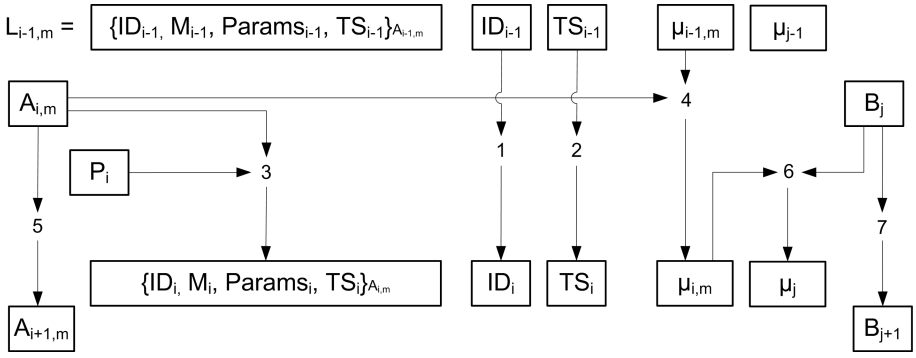


Figura 2. Concatenação de uma nova entrada $L_{i,m}$ não compactada.

1. ID_i é comparado com ID_{i-1}. Como são iguais nesse caso, ID_i é descartado e ID_{i-1} é mantido.
2. $TS_{dif} = TS_i - TS_{i-1}$, a diferença horária entre as entradas. TS_i é guardado e TS_{i-1} é apagado.
3. $L_{i,m} = E_{A_{i,m}}(P_i)$, gera a nova entrada a partir da cifragem simétrica do *payload* P_i com a chave $A_{i,m}$. Onde $P_i = (TS_{dif}, [Params_i])$.

O restante do procedimento é idêntico aos passos 4 à 7 descritos para a concatenação de uma entrada não compactada.

5.3. Verificação do log de auditoria

A verificação do log de auditoria pode ser executada tanto pelo verificador V, quanto pelo servidor T. Para iniciar o procedimento, V requisita e obtém, de forma segura, de T a chave inicial de verificador. Com a chave em posse, V consegue calcular todas as chaves iniciais de todas as categorias de log através da função *SbC-EC.Ckg*. V requisita os arquivos de log de seu interesse para U e recalcula a tag de verificador $\mu_{A_{i,m}}$, pela Equação 1, de cada arquivo de log m. V então verifica se as tags são idênticas às que U armazena. Se sim, o log de auditoria está íntegro, caso contrário, isso implica que ao menos um componente de toda a sequência não está de acordo e, portanto, não está íntegro.

Como dito anteriormente, a estratégia de separação por categorias oferece uma vantagem de que, se V estiver interessado em apenas um tipo de log (p. ex., atividades de rede), ele só precisará requisitar e verificar o arquivo de log da categoria desejada, sem a necessidade de verificar todas as entradas de todos os arquivos de log. Na verificação feita por T, é necessário requisitar e verificar todos os arquivos de log. T realiza o mesmo procedimento de recalcular as tags de verificador para cada arquivo de log e, por fim, recalcula a tag da parte confiável pela Equação 2 e compara essa tag com a tag da parte confiável informada por U. Se as duas tags forem iguais, os arquivos de log estão íntegros, caso contrário, não estão.

6. Discussão de Segurança

O esquema SbC-EC MAC utiliza o esquema *FssAgg* como camada de autenticação para criar as tags de verificador. A cada nova entrada de log gerada, o protocolo recria a tag única, e de tamanho constante, para o arquivo de log correspondente. A segurança

do esquema *FssAgg* é provada nos trabalhos de Ma e Tsudik [Ma e Tsudik 2007] [Ma e Tsudik 2008] [Ma e Tsudik 2009b] [Ma e Tsudik 2009a]. Além disso, o esquema SbC-EC MAC é resistente ao ataque de *truncation* devido à propriedade "all-or-nothing" de verificação do arquivo de log herdado da autenticação *FssAgg*.

Por possuir a chave secreta inicial A_0, um usuário/verificador, V, autorizado e mal-intencionado é capaz de realizar alterações nas entradas de log e recriar as assinaturas agregadas (i.e., as tags de verificador) com o objetivo de esconder uma manipulação feita. Outros usuários autorizados, portanto, não poderão detectar essas alterações feitas por V e verificariam com sucesso os arquivos de log. A detecção da manipulação por V é possível apenas para o servidor T com ajuda da tag da parte confiável μ_j. Dessa forma, o esquema SbC-EC MAC é suscetível ao tipo ataque denominado *delayed detection* (detecção atrasada), devido ao fato de que nenhum V pode detectar prontamente uma manipulação realizada por outro V, e que a comunicação entre U e T só acontece em períodos de tempo indeterminados.

Devido a isto, a tag da parte confiável μ_j torna-se necessária para a segurança dos arquivos de log contra o atacante com privilégios, como V. Essa tag precisa ser atualizada após cada concatenação de uma nova entrada de log, possibilitando que qualquer alteração realizada nas tags de verificador seja detectada pelo servidor de coleta. μ_j é calculada a partir de uma função $MAC(X)$ após a geração de uma nova entrada de log, onde X é a concatenação de todas as tags de verificação em um determinado tempo t. μ_j é não-falsificável para os períodos antes de t, uma vez que a chave da parte confiável, B, evolui através de uma *pseudo-random function* após cada atualização dessa tag.

Por fim, salientamos que o esquema SbC-EC MAC oferece a propriedade de *forward-secrecy* e *forward secure stream integrity*. A propriedade *forward-secrecy* provê a permanência da confidencialidade das entradas de log concatenadas ao arquivo antes de uma intrusão, ou seja, mesmo que um atacante consiga invadir o sistema no tempo t, a confidencialidade das entradas geradas antes de t é mantida. Da mesma forma, a propriedade *forward secure stream integrity* também garante a integridade da sequencia das entradas do arquivo de log antes do momento t. Essas duas propriedades tem como base o mecanismo de evolução de chaves simétricas.

7. Avaliação

Nesta seção, o esquema SbC-EC MAC é avaliado de acordo com o ganho de armazenamento em memória em relação a outros esquemas de log seguro de criptografia simétrica existentes na literatura: (i) Schneier-Kelsey e (ii) FssAgg MAC. A seguir, descrevemos algumas premissas sobre o armazenamento necessário de cada item presente nos esquemas e, por fim, uma simulação é realizada com fins comparativos.

Seja M_{key} o espaço de armazenamento necessário para guardar as chaves privadas, M_{mac} o espaço necessário para guardar uma saída obtida de uma função $MAC()$ (p. ex.,

Tabela 1. Sobrecarga da autenticação dos esquemas de log seguro

	Sobrecarga	Ordem
Schneier-Kelsey	$M_{key} + N(M_{hash} + M_{mac})$	$O(N)$
FssAgg MAC	$2M_{key} + 2M_{hash}$	$O(1)$
SbC-EC MAC	$(c+1)M_{key} + cM_{hash} + M_{mac}$	$O(c)$

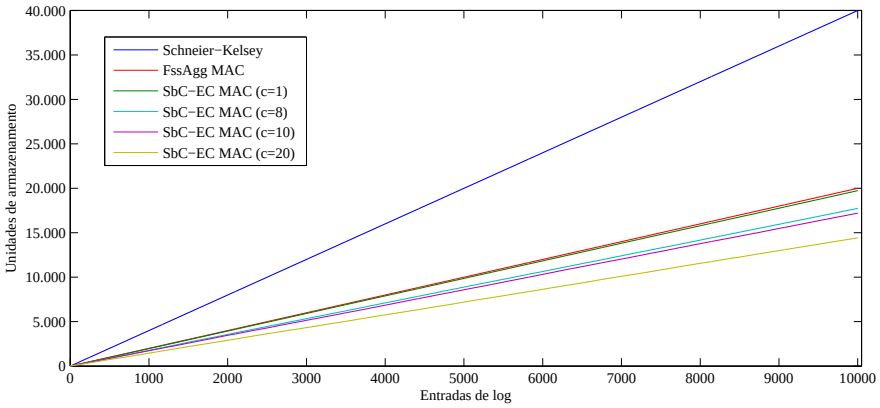


Figura 3. Simulação dos esquemas de log seguro em relação às unidades de armazenamento para N = 10.000.

assinaturas, tags) e M_{hash} o espaço necessário para guardar uma saída obtida de uma função hash $\mathcal{H}()$, dado N o número total de entradas de log geradas e c o número de categorias de log, verificamos na Tabela 1 a sobrecarga de armazenamento de autenticação associada a cada esquema. Pela tabela conclui-se que, a sobrecarga de autenticação do esquema de Schneier-Kelsey está diretamente ligada ao número de entradas logadas, uma vez que esse esquema mantém uma tag de autenticação para cada entrada. O esquema FssAgg MAC é o que implica na menor sobrecarga, utilizando apenas duas chaves e duas tags para autenticação de todo o arquivo de log. O esquema SbC-EC MAC apresenta uma sobrecarga de autenticação diretamente relacionada ao número de categorias de log estipuladas, pois, para cada categoria, uma chave e uma tag são mantidas.

Para avaliar o ganho de espaço de armazenamento do esquema SbC-EC MAC, realizamos uma simulação de geração de entradas de log a fim de calcular o tamanho total ocupado pelo log de auditoria protegido pelos esquemas testados. Para essa simulação, definimos um dispositivo capaz de registrar 400 mensagens de log diferentes (i.e. 400 IDs) e que segue uma distribuição normal para a geração de logs. Assumimos que, M_{key}, M_{mac} e M_{hash} são equivalentes a 1 unidade de armazenamento, o tamanho médio de uma entrada de log é de 2 unidades de armazenamento e o tamanho médio de uma entrada de log compactada é de 1 unidade de armazenamento. Os demais itens que compõe os logs de auditoria de cada esquema puderam ser desconsiderados, por não impactar significativamente no total de armazenamento ocupado. O gráfico da Figura 3 mostra o resultado de uma rodada da simulação de N = 10.000 entradas de log para cada esquema de log seguro em relação às unidades de armazenamento necessárias para guardar os arquivos de log.

Com o intuito de obter o tamanho médio e o ganho de armazenamento dos esquemas FssAgg MAC e SbC-EC MAC em relação ao esquema de Schneier-Kelsey, realizamos 1.000 rodadas de simulação para N = 10.000 entradas de log. Os resultados são mostrados na Tabela 2. Para os esquema de Schneier-Kelsey e FssAgg MAC, o armazenamento se mantém constante para todas as rodadas, uma vez que eles detém o mesmo comportamento independente das entradas de log geradas. O FssAgg MAC oferece um ganho de armazenamento de 50% em relação ao esquema anterior. Para o esquema SbC-EC MAC, observa-se que, quanto maior o número de categorias c estipulado, menor são as unidades de armazenamento necessárias para guardar o log de auditoria. Isso é expli-

Tabela 2. Resultado da simulação em 1.000 rodadas

	Tamanho médio (un. de armazenamento)	Desvio padrão (σ)	Ganho de arm. (%)
Schneier-Kelsey	40.000	0	-
FssAgg MAC	20.000	0	50,00
SbC-EC MAC (c =1)	19.719	16,60	50,70
SbC-EC MAC (c =8)	17.743	42,56	55,64
SbC-EC MAC (c =10)	17.181	46,42	57,05
SbC-EC MAC (c =20)	14.379	60,39	64,05

cado pelo fato de que há mais probabilidade para $ID_i = ID_{i-1}$ quando há mais divisões, agrupando entradas de log cada vez mais semelhantes. Dessa forma, nosso esquema permite um maior ganho de armazenamento em relação ao FssAgg MAC. Essa avaliação evidencia a importância da categorização de mensagens de log no sistema para o esquema SbC-EC MAC e que, quanto maior a granularidade dessa categorização, maior o ganho de armazenamento.

8. Conclusão e Trabalhos Futuros

Neste trabalho, apresentamos uma proposta de um novo esquema de estruturação de log seguro, o SbC-EC, apropriado para dispositivos com restrições de memória de armazenamento e de comunicação em rede. O esquema de estruturação SbC-EC é fundamentado em duas características com o objetivo de reduzir o espaço necessário para armazenar entradas de log: (i) Separação por Categoria (*Split by Category*, SbC) e (ii) Compactação de Entradas de log (*Entry Compaction*, EC).

Apresentamos, também, um novo esquema de log seguro baseado nessa nova estruturação, o SbC-EC MAC. O esquema SbC-EC MAC utiliza as primitivas da criptografia simétrica e o esquema de autenticação *FssAgg*, criado por Ma e Tsudik [Ma e Tsudik 2007], para criar as tags de autenticação do log de auditoria. Descrevemos as funções presentes no SbC-EC MAC, como é realizada a inicialização e verificação dos arquivos de log, a concatenação e a compactação das entradas de log. Além disso, são feitas algumas considerações sobre a segurança do esquema, onde expomos algumas características como a *forward-secrecy*, a *forward secure stream integrity* e a suscetibilidade ao ataque de *delayed detection*, causado por um usuário privilegiado. Por fim, demonstramos uma avaliação do esquema SbC-EC MAC em relação ao ganho de armazenamento, comparando-o a outros dois esquemas simétricos existentes na literatura.

Como trabalhos futuros, consideramos adicionar características ao esquema, levando em conta a criticidade da informação do log e desempenho computacional necessário para proteção dos arquivos de log. Assim como elaborar, baseado na estruturação SbC-EC, um esquema de log seguro para o domínio da criptografia de chave-pública.

Referências

Accorsi, R. (2011). BBox: A distributed secure log architecture. In *proceedings of the 7th European Workshop on Public Key Infrastructures, Services and Applications*, páginas 109–124.

Accorsi, R. (2013). A secure log architecture to support remote auditing. *Mathematical and Computer Modelling*, 57(7-8):1578–1591.

Bellare, M., Canetti, R., e Krawczyk, H. (1996). Keying hash functions for message authentication. páginas 1–15. Springer-Verlag.

Bellare, M. e Yee, B. (1997). Forward integrity for secure audit logs. Technical report, Computer Science and Engineering Department, University of California at San Diego.

Bellare, M. e Yee, B. (2003). Forward-security in private-key cryptography. *Topics in Cryptology-CT-RSA 2003*, páginas 1–24.

Holt, J. E. (2006). Logcrypt: Forward Security and Public Verification for Secure Audit Logs. In *Proceedings of the 2006 Australasian Workshops on Grid Computing and e-Research - Volume 54*, ACSW Frontiers '06, páginas 203–211, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

International Electrotechnical Commission (2011). ISA - Security for Industrial Automation and Control Systems - Technical Security Requirements for IACS Components - Part 4.

Ma, D. e Tsudik, G. (2007). Extended Abstract: Forward-Secure Sequential Aggregate Authentication. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, páginas 86–91, Washington, DC, USA. IEEE Computer Society.

Ma, D. e Tsudik, G. (2008). A New Approach to Secure Logging. In *Proceeedings of the 22Nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, páginas 48–63, Berlin, Heidelberg. Springer-Verlag.

Ma, D. e Tsudik, G. (2009a). A New Approach to Secure Logging.

Ma, D. I. e Tsudik, G. (2009b). A New Approach to Secure Logging. *ACM Transactions on Storage*, 5(1):2:1—-2:21.

National Institute of Standards and Technology (2001). Announcing the Advanced Encryption Standard (AES).

National Institute of Standards and Technology (2012). FIPS PUB 180-4, Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication 180-4. Technical report, Department Of Commerce.

Oprea, A. e Bowers, K. D. (2009). Authentic time-stamps for archival storage. In *Proceedings of the 14th European Conference on Research in Computer Security*, ESORICS'09, páginas 136–151, Berlin, Heidelberg. Springer-Verlag.

Schneier, B. e Kelsey, J. (1998). Cryptographic Support for Secure Logs on Untrusted Machines. In *Proceedings of the 7th Conference on USENIX Security Symposium - Volume 7*, SSYM'98, página 4, Berkeley, CA, USA. USENIX Association.

Schneier, B. e Kelsey, J. (1999). Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security*, 2(2):159–176.

Yavuz, A. A. e Ning, P. (2009). BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems. In *Computer Security Applications Conference, 2009. ACSAC '09. Annual*, number ii, páginas 219–228.

Yavuz, A. a., Ning, P., e Reiter, M. K. (2012). BAF and FI-BAF: Efficient and Publicly Verifiable Cryptographic Schemes for Secure Logging in Resource-Constrained Systems. *ACM Transactions on Information and System Security*, 15(2):1–28.

Detecção de Dados Suspeitos de Fraude em Organismos de Inspeção Acreditados

Rosembergue P. Souza[1], **Luiz F. R. C. Carmo** [1], **Luci Pirmez**[2]

[1]Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro)
Av N. S. das Graças, 50, 25.250-020, Xerém - Duque de Caxias - Rio de Janeiro

[2]Programa de Pós Graduação em Informática Instituto Tércio Pacitti
Instituto de Matemática, Universidade Federal do Rio de Janeiro, 21.941-901, Rio de Janeiro

{rpereira,lfrust}@inmetro.gov.br, luci.pirmez@nce.ufrj.br

Abstract. *In recent years, there have been some news reports about fraud cases in activities regulated by Inmetro. Searching for new Information Technology tools to control these activities, this paper presents a mechanism to detect suspected fraud results in vehicle safety inspections. As a solution to the problem, the Markov Decision Process has been combined with Benford's Law. With the planning power of Markov Decision Process it is possible to find a subset of data with high probability of fraud.*

Resumo. *Nos últimos anos, algumas reportagens jornalísticas apontaram casos de fraudes em serviços regulamentados pelo Inmetro. Indo ao encontro de novas ferramentas de Tecnologia da Informação para acompanhamento de atividades regulamentadas pelo Inmetro, este trabalho apresenta um mecanismo para detecção de resultados suspeitos de fraude em inspeções de segurança veicular. O problema é solucionado usando Processo de Decisão de Markov combinado com a Lei de Benford. Através do poder de planejamento do Processo de Decisão de Markov é possível selecionar um subconjunto de dados com alto grau de suspeita de fraude.*

1. Introdução

Nos últimos anos, algumas reportagens jornalísticas repercutiram nacionalmente quando apontaram casos de fraudes em serviços regulamentados pelo Inmetro. Dois exemplos ilustram essa situação: o caso de fraude em bombas de combustível e o caso de roubo de combustível durante o transporte até os postos de venda. A fraude em bombas de combustível ocorreu através de controle remoto do volume de combustível que seria fornecido ao cliente. Por sua vez, o roubo de combustível durante o transporte até os postos de venda dava-se através da instalação de dois tanques no caminhão, sendo que um deles era destinado a desviar grande quantidade do produto.

O Inmetro tem procurado estabelecer ações de melhoria no acompanhamento dos serviços regulamentados pelo órgão. Uma dessas ações é o uso intensivo de ferramentas de Tecnologia da Informação (TI). Para ilustrar, tem-se a proposta de um sistema de acompanhamento da consistência de inspeções dos organismos acreditados na área equipamentos que transportam produtos perigosos através de critérios de plausibilidade [Machado et al. 2011]. Outro exemplo é a utilização de Rede Neurais de Kohonen para

identificar as características de possíveis organismos de inspeção fraudulentos na área equipamentos que transportam produtos[Souza et al. 2013].

Indo ao encontro de novas ferramentas de TI para acompanhamento de atividades regulamentadas pelo Inmetro, neste trabalho apresenta-se um mecanismo para detecção de resultados suspeitos de fraude em inspeções de segurança veicular. Os requisitos para a realização das inspeções de segurança veicular estão dispostos nas Portarias Inmetro Nº 30/2004, Nº 32/2004 e Nº 49/2010. Uma das etapas dessa inspeção é a verificação do grau de emissão de monóxido de carbono (CO) e hidrocarboneto (HC) pelos veículos inspecionados. A emissão destes gases por automóveis segue uma distribuição estatística do tipo gama [Guo et al. 2007], e conforme os veículos vão envelhecendo e acumulando quilômetros rodados a tendência é que o grau de emissão aumente [Wenzel et al. 2000]. A intenção de se inspecionar os veículos quanto a emissão de poluentes é manter o índice de emissão dentro dos valores estabelecidos pelos órgãos ambientais.

Somando-se a isso, as Portarias Inmetro Nº 30/2004, Nº 32/2004 e Nº 49/2010 estabelecem que toda inspeção deve ser conduzida por um organismo de inspeção autorizado pela Coordenação Geral de Acreditação do Inmetro (Cgcre/Inmetro). A Cgcre/Inmetro utiliza equipes de avaliadores para autorizar e supervisionar essas empresas. Logo, uma das formas de a Cgcre/Inmetro verificar se existem anomalias nos resultados das inspeções de seus organismos acreditados é se valendo de testes estatísticos.

Um dos testes estatísticos que se pode utilizar para verificar se houve adulteração em dados numéricos é a Lei de Benford. A Lei de Benford preconiza que, em alguns conjuntos de dados, a frequência de aparição dos dígitos mais significativos de um número segue uma distribuição logarítmica [Nigrini 2012]. Algumas distribuições estatísticas como a exponencial, gama e log-normal atendem a Lei de Benford de forma aproximada [Formann 2010]. Portanto, espera-se que os resultados dos ensaios de emissões de gases poluentes encontrados pelos organismos acreditados sigam de forma aproximada a Lei de Benford. No entanto, apenas o uso da Lei de Benford não é suficiente para confirmação de casos de fraude, mais investigações são sempre necessárias para se chegar a conclusão de fraude [Nigrini 2012]. Por isso, neste trabalho, para refinar os resultados da aplicação da Lei de Benford utiliza-se a técnica de Processo de Decisão de Markov - MDP (*Markov Decision Process*).

O Processo de Decisão de Markov é um problema de decisão sequencial para um ambiente completamente observável, estocástico, com um modelo de transição de Markov e recompensas aditivas [Russell and Norvig 2013]. A investigação realizada por auditores para verificar se a prestação de um serviço atende ou não a uma norma encaixa-se num problema de MDP. Um auditor, representando o agente do MDP, analisa registros em uma auditoria de forma sequencial. A cada novo registro analisado, o auditor decide qual fará parte de seu conjunto de evidências de atendimento a uma norma. Geralmente, não há tempo para analisar todos os registros produzidos por uma empresa, assim ele escolhe um subconjunto desses registros, onde cada registro possui uma probabilidade de ser escolhido.

Além disso, os registros coletados são ligados pelos seus atributos em comum, sendo que a escolha do próximo registro depende dos atributos do registro atual. Para aplicação de um MDP considerar-se-á que a escolha do próximo registro depende apenas

do atributos do registro atual. Registros anteriores não terão influência na decisão do próximo registro a compor um caso de suspeita de fraude. Com isso se estabelece a propriedade de Markov necessária a aplicação de um MDP. O objetivo da aplicação de um MDP é utilizar seu poder de planejamento para selecionar um subconjunto de dados com maior grau de suspeita de fraude.

Por fim, este trabalho está organizado da seguinte maneira: a seção 2 descreve os os trabalhos relacionados, em seguida a seção 3 mostra alguns conceitos básicos, a seção 4 discorre sobre a proposta do trabalho, a seção 5 apresenta o estudo de caso e por sua vez a seção 6 mostra as conclusões e trabalhos futuros .

2. Trabalhos Relacionados

Existem trabalhos na literatura em que a Lei de Benford foi combinada com alguma técnica de inteligência artificial com o intuito de potencializar os resultados dessa lei.

Em [Bhattacharya et al. 2011], utiliza-se uma rede neural otimizada por algoritmos genéticos para classificar se um determinado conjunto de dados possui conformidade com a Lei de Benford ou não. Nesse trabalho, os autores conseguem, através da rede neural, unir testes estatísticos, medidas de teoria da informação e o coeficiente de correlação de Pearson para decidir sobre o atendimento a tal lei. Embora tenha sido um avanço na área, mais trabalho de investigação ainda é necessário para verificar quais são os elementos que estão causando os desvios identificados.

Em [Cantu and Saiegh 2011], a Lei de Benford é utilizada em conjunto com a técnica de Naive Bayes para classificação de eleições presidenciais quanto a sua legitimidade. Os autores utilizam a Lei de Benford para criação de dados sintéticos sobre eleições legítimas e fraudulentas, dada a dificuldade de se encontrar bancos de dados com exemplos desses tipos de eleições. Esses dados sintéticos são utilizados para treinamento da técnica de Naive Bayes. A validação do trabalho utiliza as eleições ilegítimas ocorridas na Argentina no período de 1931-1941, considerada a década infame desse país. A abordagem adotada permite a utilização direta dos desvios da Lei Benford para classificação da legitimidade de uma eleição, não havendo a necessidade de se verificar os elementos que causaram o não atendimento a essa lei.

Em [Lu 2007], usa-se a Lei de Benford com a técnica de Aprendizagem por Reforço. O intuito é aproveitar a característica exploratória da Aprendizagem por Reforço para conectar atributos de um conjunto de dados com alto grau de anomalia. A técnica de Aprendizagem por Reforço é similar a um Processo de Decisão de Markov, a diferença consiste no fato de que na Aprendizagem por Reforço não se conhece a função de probabilidades de transição entre estados, nem a função de recompensa. O agente descobre a política ótima para um determinado ambiente através de tentativa e erro. Essa combinação de técnicas permite o refinamento dos resultados obtidos pela Lei de Benford. No entanto, a necessidade de recurso computacional aumenta a medida que o número de dados a serem analisados aumenta.

Assim, neste trabalho, combina-se a Lei de Benford com a técnica de Processo de Decisão de Markov para se montar um caso de fraude. De forma similar a [Lu 2007], a intenção é conectar atributos dos elementos de um conjunto maximizando o grau de anomalia entre eles. Para isso, o conjunto de dados disponível é modelado nos elementos

que compõem um MDP. Esta modelagem permite o uso de programação dinâmica para se encontrar uma política ótima. A obtenção dessa política ótima é feita de forma mais rápida usando-se MDP do que Aprendizagem por Reforço, visto que não se precisa visitar os estados do ambiente um número infinito de vezes para se chegar ao ponto ótimo.

3. Fundamentação

3.1. Lei de Benford

Segundo a Lei de Benford, em alguns conjuntos de dados, a frequência de aparição dos dígitos mais significativos segue uma distribuição logarítmica [Nigrini 2012]. As equações 1 e 2 apresentam as fórmulas das probabilidades esperadas para o primeiro e para os dois primeiros dígitos mais significativos respectivamente:

$$P(D_1 = d_1) = \log\left(1 + \frac{1}{d_1}\right) \tag{1}$$

$$P(D_1 D_2 = d_1 d_2) = \log\left(1 + \frac{1}{d_1 d_2}\right) \tag{2}$$

onde D_1, representa o primeiro dígito mais significativo, $D_1 D_2$ representa os dois primeiros dígitos mais significativos, $d_1 \in \{1, 2, 3, ...9\}$ e $d_1 d_2 \in \{10, 11, 12, 13, ..., 99\}$.

Segundo [Nigrini 2012], a Lei de Benford só pode ser aplicada a uma amostra de dados se as seguintes condições forem satisfeitas: i) os dados dessa amostra devem conter informação de tamanho de fatos ou eventos. Por exemplo, tamanho de cidades, vazão de rios e lucro de empresas; ii) a amostra não deve possuir mínimos e máximos embutidos, exemplo: um fundo de investimento com valor mínimo de R$500,00 de aplicação; iii) os elementos da amostra não podem ser dados de identificação, como número de telefone e placas de veículos; iv) a média dos dados deve ser menor que a mediana e os dados não devem ficar fortemente agrupados em torno do valor médio.

3.2. Processos de Decisão de Markov

Um Processo de Decisão de Markov é uma tupla (S, A, T, R) onde: S é o conjunto de estados , A é o conjunto de ações , $T : S \times A \times S \rightarrow [0, 1]$ é uma função de probabilidade de transição do estado $s \in S$ para $s\prime \in S$, dado uma ação $a \in A$ (denotada por $T(s\prime \mid s, a)$) e $R : S \times A \rightarrow \mathbb{R}$ é uma função que dá o custo (ou recompensa) quando o agente está no estado $s \in S$ toma uma decisão $a \in A$ e vai para o estado $s\prime \in S$ (denotada por $R(s\prime \mid s, a)$) [David and Alan 2010]. O nome Markov se deve a propriedade Markoviana (sem memória), isto é a definição do próximo estado do agente só depende do estado atual. Uma política π é uma função que mapeia estados em ações, sendo que o objetivo do Processo de Decisão de Markov é encontrar uma política que maximize sua recompensa acumulada ao longo prazo. Uma forma de se medir o desempenho do agente num MDP é usando o critério de recompensa esperada descontada $E\left[\sum_{k=0}^{\infty} \gamma^k r_k\right]$, onde r_k é a recompensa no passo k e γ é o fator de desconto, que é usado para garantir a convergência do valor da recompensa total esperada.

A função $V^{\pi}(s)$ é o valor esperado da recompensa descontada para o agente que sai do estado s e segue a política π. Já a função $Q^{\pi}(s, a)$ é o valor da recompensa esperada

descontada quando o agente sai do estado s escolhendo a ação a e seguindo a política π. A função de valor $V^*(s)$ ótima é definida como $V^*(s) = \underset{\pi}{max}(V(s))$ para todo $s \in S$. Valendo também $V^*(s) = \underset{a}{max}(Q^*(s,a))$ e $\pi^* = \underset{a}{argmax}(Q^*(s,a))$. Existe uma grande quantidade de algoritmos para a solução de um MDP. Alguns trabalham diretamente com políticas, enquanto outros trabalham com funções valor, detalhes sobre esses algoritmos podem ser encontrados em [David and Alan 2010].

4. Proposta

O mecanismo de detecção de dados suspeitos de fraude proposto neste trabalho inicia-se aplicando a Lei de Benford na amostra de dados. A Lei de Benford é utilizada para determinar a função de recompensa do MDP. Tendo em mãos a função de recompensa, a tabela de dados da amostra passa por um processo de discretização dos atributos. Em seguida, define-se a função de probabilidade de transição entre estados. Daí, uma política ótima é estabelecida. Com esta política ótima, explora-se a amostra de dados e apresenta-se uma lista com os casos de suspeita de fraude. A figura 1 mostra as etapas do processo para detecção de dados suspeitos de fraudes usando MDP.

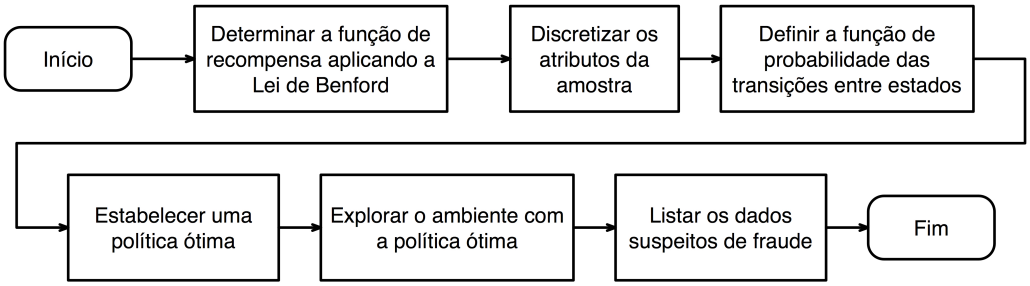


Figura 1. Representação gráfica da detecção de dados suspeitos de fraude usando Processo de Decisão de Markov.

4.1. Definindo a função de recompensa usando a Lei de Benford

Para que se possa aplicar o Processo de Decisão de Markov, faz-se necessário definir a função de recompensa $R(s\prime \mid s, a)$. A fim de facilitar a compreensão dessa definição, é preciso caracterizar o conjunto de ações e de estados do MDP.

Primeiramente usa-se uma representação tabular, onde cada linha da tabela representa um elemento da amostra de dados e cada coluna representa um atributo desse elemento. As linhas são o conjunto de estados possíveis S, onde cada linha é um estado s. Os atributos são o conjunto de ações A, sendo cada atributo uma ação a . A tabela 1 ilustra como são representados os estados e ações. No exemplo mostrado nessa tabela, os dados contém informações sobre ensaios de emissões veiculares de gases poluentes. Os atributos são Vel velocidade de rotação do motor em marcha alta em rpm na hora do ensaio, $Fdil$ fator de diluição dos gases do ensaio, CO índice percentual de emissão de monóxido de carbono, $CO2$ índice percentual de emissão de dióxido de carbono e HC índice de emissão de Hidrocarboneto em ppm.

A recompensa de cada estado é associada com o grau de anomalia daquele estado. Neste trabalho, para definir esse grau de anomalia usou-se a Lei de Benford. A tabela 1 ilustra a forma como as recompensas foram definidas . Para aplicação da Lei de Benford considerou-se apenas os dados de emissão de HC, denominado, neste trabalho, como indicação "Alvo".

Tabela 1. Ilustração do conjunto de estados, ações e recompensas

	Atributos					
	Ações				Alvo	
Estados	Vel (rpm)	Fdil	CO(%)	CO2(%)	HC(ppm)	Recompensa
1	2571	1,67	0,13	8,83	42	0,84
2	2479	1,12	0,44	12,83	129	0,98
3	2547	1,00	0,01	13,51	16	1,19
4	2426	1,11	0,00	13,49	22	1,18

Como dito anteriormente, os valores de HC seguem uma distribuição estatística gama, logo espera-se que esses valores atendam a Lei de Benford de forma aproximada. Assim, o grau de anomalia de cada estado é determinado pelo grau de desvio que os valores de HC têm perante a Lei de Benford. O grau de anomalia de cada estado s é calculado usando equação 3 descrita abaixo

$$Grau\ anomalia(s) = \frac{Prob_{observada}(s)}{Prob_{esperada}(s)} \tag{3}$$

onde $Prob_{observada}$ é a frequência observada dos dígitos mais significativos na amostra de dados e $Prob_{esperada}$ é a frequência esperada dos dígitos mais significativos segundo a Lei de Benford [Lu 2007]. A equação 4 define a função de recompensa do MDP

$$R(s\prime \mid s, a) = Grau\ anomalia(s\prime) = \frac{Prob_{observada}(s\prime)}{Prob_{esperada}(s\prime)} \tag{4}$$

Por exemplo, se forem usados os dois dígitos mais significativos dos elementos de uma amostra e considerando o valor de HC no estado 1 da tabela 1 , tem-se $D_1 = 4, D_2 = 2$. Suponha que essa combinação dos dois dígitos mais significativos apareça 12 vezes numa amostra de dados de HC com 1400 elementos, então se teria $Prob_{observada} = 12/1400 = 0,0086$. A probabilidade esperada para os dois dígitos mais significativos quando $D_1 = 4, D_2 = 2$ é, segundo a equação 2, $Prob_{esperada} = 0,0102$. Portanto, o grau de anomalia/recompensa do estado 1 é 0,84.

4.2. Discretização dos dados

Com o intuito de se aplicar a técnica de Processo de Decisão de Markov, fez-se necessário discretizar os atributos da amostra. Quando os elementos da amostra não possuem classes atribuídas a eles, deve-se se lançar mão de técnicas de discretização não-supervisionadas.

Na literatura, existem duas técnicas bastante simples para discretização não-supervisionada, a primeira estabelece intervalos com larguras iguais e a segunda estabelece um número uniforme de elementos por intervalo. O número de intervalos N_{int} é escolhido *a priori* em ambas as técnicas [Garcia et al. 2013]. Neste trabalho, adotou-se a segunda técnica de discretização, pois a primeira pode retornar intervalos muito populosos e outros intervalos com poucos elementos, deixando o resultado da discretização enviesado. A tabela 2 mostra um exemplo de atributos discretizados, supondo um número de 200 intervalos. A coluna alvo não é mostrada, pois já foi utilizada para a determinação das recompensas.

4.3. Definindo a função transição de probabilidades T

A solução de um MDP consiste em estabelecer uma política ótima levando em conta a tupla (S, A, T, R). Até o momento, já foram modelados o conjunto de estados S, o conjunto de ações A e a função de recompensa R. O próximo passo é definir a função de probabilidades T.

Para se definir a função T, primeiro deve-se ter em mente que os estados são conectados através de atributos em comum. Considerando o método de discretização utilizado, onde o número de elementos N_e por intervalo é uniforme, apenas elementos de mesmo intervalo podem formar estados conectados. Admitindo transições equiprováveis, a probabilidade do agente sair de um estado s para um estado $s\prime$ escolhendo uma determinada ação a será $\frac{1}{N_e}$. Como o número de elementos de cada intervalo $N_e = N_s/N_{int}$, então a probabilidade de transição de um estado para o outro é $\frac{1}{N_e} = \frac{N_{int}}{N_s}$. Assim, a função de transição $T(s\prime \mid s, a)$ pode ser definida pela equação

$$T(s\prime \mid s, a) = \frac{N_{int}}{N_s} \tag{5}$$

onde N_s é o número total de estados.

Tabela 2. Atributos com valores discretizados

	Atributos/Ações				
Estados	Vel (rpm)	Fdil	CO(%)	CO2(%)	Recompensa
5	110	1	166	91	1,41
6	126	173	158	31	1,22
7	9	1	167	94	0,49
8	135	1	160	41	1,11

Para exemplificar como as transições entre estados acontecem, considere que o agente só pode transitar entre os estados mostrados na tabela 2. A figura 2 mostra o diagrama de transições da mudança de estados que o agente pode realizar nessa situação. Suponha que o agente está no estado 5 e escolhe a ação $Fdil$, o conteúdo de $Fdil$ no estado 5 é 1, logo os estados 7 e 8 estão conectados a 5. Uma vez escolhida a ação $Fdil$ no estado 5, e se verificando os estados conectados a esse estado, a decisão do próximo estado do agente será feita através de um sorteio. Cada próximo estado tem

a mesma probabilidade de ser sorteado. Inclui-se nesse sorteio também o estado 5, já que ele próprio possui o conteúdo 1. Isso significa que o agente pode tomar uma ação e permanecer em seu estado atual. Assumindo transições equiprováveis, a probabilidade de transição entre os estados 5, 7 e 8 é de 1/3. Caso no sorteio o agente saia do estado 5 e cai no estado 8, ele recebe a recompensa $1,11$. Isto é representado na figura 2 pela tupla (1; 0,333; 1,11), onde o primeiro elemento representa o valor da ação escolhida, o segundo representa a probabilidade da transição e o terceiro, a recompensa a ser recebida ao final da transição.

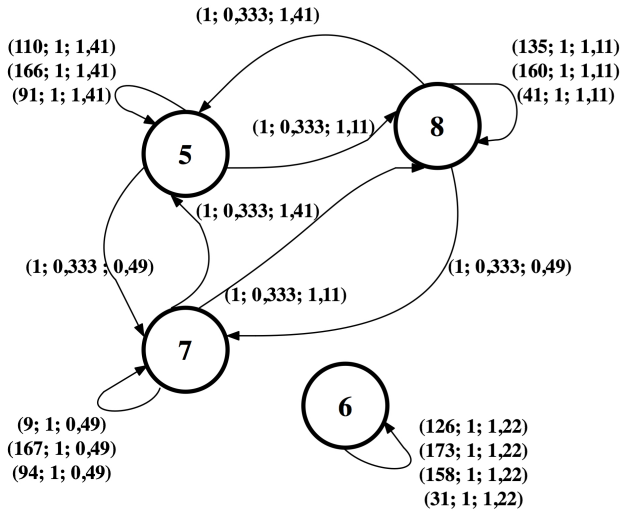


Figura 2. Representação gráfica do diagrama de transições.

4.4. Estabelecendo uma política ótima

Uma vez que a tupla (S, A, T, R) da MDP está definida, pode-se utilizar o algoritmo 1 para se encontrar uma política ótima.

O algoritmo 1 é chamado de iteração de valor assíncrona, pois o cálculo da função $Q(s, a)$ para cada par estado-ação é realizado em qualquer ordem. Além disso, vale destacar que o critério de parada indica que a diferença absoluta entre a função de valor $V(s)$ e a função de valor ótima $V^*(s)$ é menor que ϵ para todo s. Por fim, o algoritmo retorna a função $Q^*(s, a)$ para que seja obtida uma política sub-ótima. Este tipo de política será utilizada posteriormente em experimentos abordados neste trabalho.

4.5. Explorando o ambiente com a política ótima

De posse da política ótima, o agente deve explorar a amostra de dados para listar os elementos suspeitos de fraude. De forma similar a [Lu 2007], a exploração ocorrerá em episódios. Um episódio é o trajeto do agente do estado inicial ao estado final. O estado inicial considerado neste trabalho é o estado cuja função de valor de estado é máxima, o estado final será qualquer estado previamente visitado.

4.6. Lista dos dados suspeitos de fraude

O agente, usando a política ótima obtida, visitará vários estados dentro da amostra de dados considerada. A lista de elementos suspeitos de fraude são todos esses estados visitados.

Algoritmo. 1 Iteração de Valor Assíncrona$(S, A, R, T, \gamma, \epsilon)$. Algoritmo para encontrar uma política ótima de um MDP.

início
inicia $\pi(s)$ arbitrariamente
inicia $Q(s, a)$ arbitrariamente
inicia $V_k(s)$ com zeros
$k := 0$
    repita
    $k := k + 1$
    seleciona randomicamente estado s
    seleciona randomicamente ação a
    $Q(s, a) := \sum_{s\prime} T(s\prime \mid s, a)(R(s\prime \mid s, a) + \gamma \underset{a\prime}{max}\ Q(s\prime, a\prime))$
    $V_{k-1}(s) := V_k(s)$
    $V_k(s) := \underset{a}{max}\ Q(s, a)$
    até $\forall s \mid V_k(s) - V_{k-1}(s) \mid < \frac{\epsilon(1-\gamma)}{\gamma}$
    para cada estado s **faça**
        $\pi^*(s) := \underset{a}{argmax}\ Q^*(s, a)$
    retorna $\pi^*, Q^*(s, a)$
fim

5. Resultados

O desempenho do mecanismo proposto neste trabalho foi medido através de 4 experimentos. Esses experimentos foram elaborados usando linguagem C do programa MATLAB®. Tais testes foram utilizados para verificar a capacidade do mecanismo em determinar um subconjunto de elementos onde se tenha o máximo grau de suspeita de fraude. Para isso, tomou-se uma amostra de 1400 ensaios de emissão veicular de gases poluentes obtidos num organismo acreditado e tabulados numa planilha Excel®. Nessa amostra, escolheu-se aleatoriamente uma quantidade predeterminada de ensaios. Nesses ensaios escolhidos aleatoriamente, foi alterada a medida de HC por um valor constante qualquer. No fim, aplicou-se o mecanismo proposto neste trabalho para verificar quantos elementos alterados eram detectados. A equação 6 mostra a métrica utilizada na medição do desempenho

$$TA_{med} = \frac{N_{alt}}{N_{rec}} \tag{6}$$

onde TA_{med} é a taxa média de acerto, N_{alt} é o número de elementos alterados e N_{rec} é o número de elementos recomendo pelo mecanismo proposto neste trabalho como sendo os mais anômalos. A média é calculada por episódio de exploração. A figura 3 mostra parte do conjunto de dados utilizados nos experimentos deste trabalho. Os atributos Vel, $Fdil$, CO, $CO2$ e HC são como explicados anteriormente.

Em todos os experimentos considera-se $\gamma = 0,9$, $\epsilon = 0,001$ e 121 episódios de exploração. Esse número de episódios de exploração foi escolhido para que se tivesse num

F	G	H	I	J
Vel (rpm)	FDIL	CO (%)	CO2(%)	HC(ppm)
2589	1,09781033	0,0973125	13,56625	34,5625
2431	1,09901684	0,0754375	13,573125	40,9375
2574	1,11591575	0	13,441875	28,375
2414	1,12238694	0	13,364375	26,8125
2666	1,11198113	0,0500625	13,439375	55,6875
2475	1,10639867	0	13,5575	49,75
2571	1,67256711	0,13575	8,8325	42

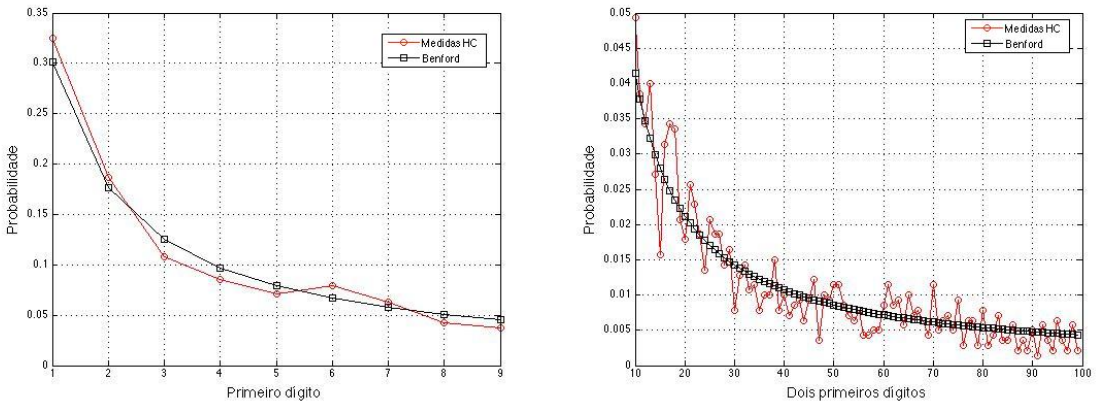
Figura 3. Parte dos dados utilizados nos experimentos deste trabalho.

número de graus de liberdade de 120. Com esse número de graus de liberdade, calculou-se os limites do intervalo de confiança da taxa média de acerto usando a distribuição estatística t e intervalo de confiança de 5%.

5.1. Experimento 1

Nesse primeiro experimento, o intuito é investigar de que forma a definição da função de recompensa pode influenciar na taxa média de acerto do mecanismo proposto neste trabalho. Duas opções foram utilizadas para definição da função de recompensa. A primeira opção considerou o grau de anomalia dado pela Lei de Benford usando apenas o primeiro dígito mais significativo. A segunda opção considerou os dois primeiros dígitos mais significativos.

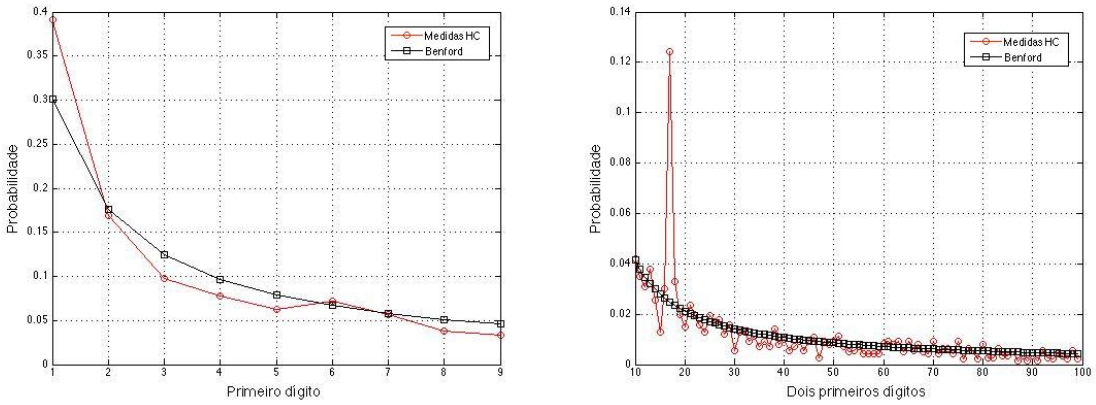
As figuras 4 e 5 mostram os resultados obtidos a partir da aplicação da Lei de Benford ao conjunto de valores de HC. A figura 4 mostra a aplicação da Lei de Benford considerando que os dados não sofreram nenhum tipo de adulteração. Já a figura 5 mostra a aplicação da Lei de Benford quando 140 valores de HC foram escolhidos aleatoriamente e adulterados para o valor de 17 ppm. Nota-se na figura 5(a) que o dígito 1 apresenta o maior grau de anomalia. Na figura 5(b), o número 17 apresenta maior grau de anomalia. Ambas as figuras 5(a) e 5(b) indicam sinais de adulteração dos dados de HC.



(a) Lei de Benford com o primeiro dígito mais significativo

(b) Lei de Benford com os dois primeiros dígitos mais significativo

Figura 4. Aplicação da Lei de Benford no conjunto de valores de HC considerando o primeiro dígito mais significativo e os dois primeiros dígitos mais significativos sem adulterações.



(a) Lei de Benford com o primeiro dígito mais significativo

(b) Lei de Benford com os dois primeiros dígitos mais significativo

Figura 5. Aplicação da Lei de Benford considerando o primeiro dígito mais significativo e os dois primeiros dígitos mais significativos. No conjunto de valores de HC há 140 valores adulterados para 17 ppm.

Tabela 3. Desempenho considerando primeiro e segundo dígitos

Dígito mais significativo	Taxa média de acerto
Primeiro	$0,00 \pm 0,00$
Dois primeiros	$0,46 \pm 0,03$

Considerando o grupos de dados adulterados, aplica-se o MDP afim de se identificar quais elementos foram adulterados. Usou-se o processo de discretização de atributos com número de intervalos $N_{int} = 200$. A tabela 3 mostra os valores da taxa média de acerto.

Observa-se que usando os dois dígitos mais significativos pôde-se identificar mais elementos adulterados. O uso da Lei de Benford considerando os dois dígitos mais significativos permitiu focar as maiores recompensas nos valores que possuíam 1 e 7 como os dois primeiros dígitos mais significativos. Isso aumentou a chance de se encontrar mais elementos adulterados.

5.2. Experimento 2

Nesse segundo experimento, investiga-se a forma como a definição do número de intervalos pode influenciar na taxa média de acerto do mecanismo proposto neste trabalho. Para tal, tomou-se os seguintes números de intervalos $N_{int} = 100$, $N_{int} = 200$ e $N_{int} = 280$. A função de recompensa foi definida pela Lei de Benford usando os dois primeiros dígitos mais significativos. Considera-se ainda que o número de elementos adulterados é de 140 valores, todos adulterados para o valor de 17 ppm de forma aleatória. A tabela 4 mostra os valores da taxa média de acerto para cada valor de N_{int}.

O número de intervalos tem influência direta na função de probabilidade das transições entre estados, bem como no número de estados que podem ser conectados entre si. Nota-se na tabela 4 que, aumentando o número de intervalos, tem-se um melhora na

Tabela 4. Desempenho considerando o número de intervalos

Intervalos	Taxa média de acerto
100	$0,23 \pm 0,03$
200	$0,46 \pm 0,03$
280	$0,67 \pm 0,04$
350	$0,18 \pm 0,03$

taxa média de acerto. No entanto, aumentar excessivamente o número de intervalos diminui a capacidade de acerto, já que diminui o número de estados que podem ser conectados entre si.

5.3. Experimento 3

Nesse terceiro experimento, verifica-se o efeito de se ter diferentes números de valores de HC adulterados. A função de recompensa foi definida pela Lei de Benford usando os dois primeiros dígitos mais significativos. Considera-se número de intervalos discretizados $N_{int} = 200$. Os valores de HC foram adulterados para o valor de 17 ppm de forma aleatória.

Tabela 5. Desempenho considerando número de elementos alterados

Número de alterações de HC	Taxa média de acerto
14	$0,00 \pm 0,00$
60	$0,00 \pm 0,00$
140	$0,46 \pm 0,03$
280	$0,47 \pm 0,04$
320	$0,69 \pm 0,04$

A tabela 5 mostra a taxa média de acerto quando o número de valores adulterados varia. Nota-se nessa tabela que quanto mais dados adulterados mais o mecanismo proposto neste trabalho consegue acertar. No entanto, é importante destacar que a Lei de Benford possui baixa sensibilidade. Quando poucos dados são modificados, a técnica não consegue destacar essas adulterações.

5.4. Experimento 4

Por fim, neste quarto experimento, investiga-se a influência de se usar uma política ótima e outra sub-ótima para se listar os dados suspeitos de adulteração. A função de recompensa foi definida pela Lei de Benford usando os dois primeiros dígitos mais significativos. Considera-se número de intervalos discretizados $N_{int} = 200$. Os valores de HC foram adulterados para o valor de 17 ppm de forma aleatória. O número de valores adulterados foi de 140.

A tabela 6 mostra a taxa média de acerto quando se usa a política ótima e quando se usa a política sub-ótima. Como era de se esperar, usando a política ótima o agente consegue encontrar mais valores adulterados do que usando a política sub-ótima.

Tabela 6. Desempenho considerando política ótima e subótima

Política	Taxa média de acerto
Ótima	$0,46 \pm 0,03$
Sub-ótima	$0,11 \pm 0,02$

Por ilustração, a figura 6 mostra um episódio de exploração da amostra de dados e a lista de dados suspeitos de adulteração. Nesse exemplo, o mecanismo proposto neste trabalho recomenda quatro ensaios como adulterados, e consegue acertar os quatro.

```
Episódio( 20.000000 ).
( 182.00 ) - Vel(rpm) - 2441.00 Fdil - 1.17 CO - 0.36 - CO2 - 12.47 HC - 17.00
 ( 321.00 ) - Vel(rpm) - 2513.00 Fdil - 1.17 CO - 0.72 - CO2 - 12.11 HC - 17.00
 ( 770.00 ) - Vel(rpm) - 2374.00 Fdil - 1.17 CO - 0.00 - CO2 - 12.83 HC - 17.00
 ( 919.00 ) - Vel(rpm) - 2483.00 Fdil - 1.17 CO - 0.86 - CO2 - 11.97 HC - 17.00
```

Figura 6. Lista de ensaios suspeitos de fraude com política ótima, 4 recomendações e 4 acertos.

Já a figura 7 mostra um exemplo de episódio onde a lista de dados suspeitos de adulteração é dada pelo uso da política sub-ótima. Embora liste mais valores como sendo suspeitos, só um deles foi de fato adulterado.

```
Episódio( 120.000000 ).
( 1.00 ) - Vel(rpm) - 2589.00 Fdil - 1.10 CO - 0.10 - CO2 - 13.57 HC - 34.56
 ( 190.00 ) - Vel(rpm) - 2340.00 Fdil - 1.09 CO - 0.00 - CO2 - 13.79 HC - 18.50
 ( 208.00 ) - Vel(rpm) - 2463.00 Fdil - 1.10 CO - 0.00 - CO2 - 13.70 HC - 63.69
 ( 414.00 ) - Vel(rpm) - 2526.00 Fdil - 1.09 CO - 0.02 - CO2 - 13.68 HC - 14.75
 ( 663.00 ) - Vel(rpm) - 2534.00 Fdil - 1.10 CO - 0.13 - CO2 - 13.57 HC - 17.00
```

Figura 7. Lista de ensaios suspeitos de fraude com política sub-ótima, 5 recomendações e 1 acerto.

6. Conclusões

Neste trabalho, foi apresentada uma aplicação de MDP para selecionar um subconjunto de dados com suspeita de fraude. Este tipo de mecanismo pode auxiliar os auditores da Cgcre/Inmetro a identificar anomalias durante as avaliações de supervisão dos organismos acreditados em segurança veicular. Mostrou-se que o uso da Lei de Benford considerando os dois dígitos mais significativos foi eficaz na identificação elementos adulterados. Além disso, apresentou-se que o número de intervalos de discretização pode influenciar na capacidade de acerto do mecanismo proposto. Somando-se a isso, constatou-se que a Lei de Benford possui baixa sensibilidade. Um número pequeno de dados adulterados não pode ser identificado por essa técnica. Para trabalhos futuros, pretende-se explorar outras técnicas estatísticas para determinação do grau de anomalia, outras formas de adulteração de dados, formas de identificação das causas das anomalias e os itens adulterados.

Referências

Bhattacharya, S., Xu, D., and Kumar, K. (2011). An ann-based auditor decision support system using benford's law. In *Decision Support Systems*, volume 50, pages 576–584. Elsevier B.V.

Cantu, F. and Saiegh, S., M. (2011). Fraudulent Democracy? An Analysis of Argentina's Infamous Decade Using Supervised Machine Learning. In *Political Analysis*, volume 19, pages 409–433.

David, L. P. and Alan, K. M. (2010). *Artificial Intelligence Foundations of Computational Agents*. Cambridge University Press, 1st edition.

Formann, A., K. (2010). The Newcomb-Benford law in its relation to some common distributions. In *PloS one*, volume 5.

Garcia, S., Luengo, J., and Sáez, J. (2013). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. In *IEEE Transactions on Knowledge and Data Engineering*, volume 25, pages 734–750.

Guo, H., Zhang, Q., Shi, Y., and Wang, D. (2007). On-road remote sensing measurements and fuel-based motor vehicle emission inventory in hangzhou, china. In *Atmospheric Environment*, volume 41, pages 3095–3107.

Lu, F. (2007). Uncovering Fraud in Direct Marketing Data with a Fraud Auditing Case Builder. In *Lecture Notes in Computer Science 4702*, pages 540–547.

Machado, R., C., Boccardo, D., R., Carmo, L., F. R. C., Prado, C., B., Nascimento, Tiago., M., Ribeiro, L., C., and Oliveira, T., D. (2011). Sistema de acompanhamento de inspeções de produtos perigosos. In *Anais do VI Congresso Brasileiro de Metrologia*.

Nigrini, M. (2012). *Benford's Law Applications for Forensic Accounting, Auditing, and Fraud Detection*. John Wiley & Sons, 1st edition.

Russell, S. and Norvig, P. (2013). *Artificial Intelligence A Modern Approach*. Elsevier, 3nd edition.

Souza, R., Carmo, L., F. R. C., Boccardo, D., R., Pirmez, L., and Machado, R., C. (2013). Redes de kohonen para detecção de fraudes em inspeções na área de transporte de produtos perigosos. In *Anais do VII Congresso Brasileiro de Metrologia*.

Wenzel, T., Singer, B., C., and Slott, R. (2000). Some issues in the statistical analysis of vehicle emissions. In *Journal of Transportation Statistics*, pages 1–14.

Segurança no Sensoriamento e Aquisição de Dados de Testes de Impacto Veiculares

Wilson S. Melo Jr[1,2], Luiz F. R. C. Carmo[1,2], Charles Prado[1], Paulo R. Nascimento[1], Luci Pirmez[2]

[1]Instituto de Metrologia, Qualidade e Tecnologia (Inmetro), RJ – Brasil

[2]PPGI iNCE/IM, Universidade Federal do Rio de Janeiro (UFRJ), RJ – Brasil

{wsjunior,lfrust,cbprado,prnascimento}@inmetro.gov.br,
luci.pirmez@ufrj.br

Abstract. *This paper deals with the cyber security of vehicular impact tests (crash tests). A model attack describes main attacks that can be launched against the sensing and data acquisition system. For each attack, it is described countermeasures based on methodologies already consolidated in the literature. However, one of the described attacks is related with the difficult on identify that the expected sensors were indeed inside the vehicle during the test. For this attack is presented an original idea which is the emission of a unique identifier for each sensor using a light device. The identifier is recovered using a camera already present in the test context. The idea feasibility is demonstrated on a practical experiment.*

Resumo. *Este trabalho trata da segurança cibernética em um teste de impacto de veículos (crash tests). Um modelo de ataque descreve os principais ataques que podem ser lançados contra os sistemas de sensoriamento e aquisição de dados. Para cada ataque, são apresentadas contramedidas baseadas em metodologias já consolidadas na literatura. Todavia, um desses ataques está relacionado com a dificuldade de se confirmar que os sensores foram de fato embarcados no veiculo durante o teste. Para este ataque é apresentada uma ideia original de emissão de um identificador único para cada sensor usando um dispositivo luminoso e a recuperação deste por uma câmera já utilizada no teste. A factibilidade da ideia é demonstrada por meio de um experimento.*

1. Introdução

Estima-se que no Brasil, a cada 11 minutos, uma pessoa morre em acidente de trânsito. Anualmente este número ultrapassa o total de 43 mil mortos e 150 mil feridos, um número de vítimas maior do que em muitos conflitos armados. Se forem calculadas as perdas sociais e econômicas, o valor estimado ultrapassa R$ 30 bilhões [Bacchieri e Barros 2011]. Dada a seriedade do problema, diversas medidas são continuamente propostas tanto na tentativa de reduzir o número de acidentes quanto de minimizar seus impactos. Neste segundo grupo, alternativas tecnológicas que vão desde o uso de materiais mais eficientes na absorção do impacto até os modernos sistemas de evasão de colisão estão disponíveis como arsenal para a indústria automotiva [Caveney 2010]. Em alguns casos, essas tecnologias tornam-se elementos de segurança obrigatórios, como é o caso recente dos *airbags* e freios ABS no Brasil. Entretanto, tais inovações resultam

no aumento do custo do veículo, de modo a gerar conflitos de interesses de mercado por parte da indústria, dos consumidores e das autoridades interessadas. Em muitos casos, a eficiência destas soluções não corresponde àquela divulgada pelo fabricante, o que torna necessária a verificação das mesmas em testes realizados por uma terceira parte.

Nesse contexto, os testes de impacto ou colisão, popularmente chamados de *crash tests*, possuem uma função consolidada em diversos países desenvolvidos e começam a ganhar destaque também nos países em desenvolvimento [Paine and Haley 2008]. No Brasil, só recentemente alguns veículos passaram a ser submetidos a testes de impacto. Os resultados dos primeiros testes são preocupantes, pois apontam os carros brasileiros como pouco seguros quando comparados ao mercado global. Em resposta, o governo brasileiro determinou que o Inmetro passe a realizar testes de impacto a partir de 2015 em seu futuro Centro de Tecnologia Automotivo, atualmente em construção.

Um teste de impacto se propõe a avaliar o comportamento dinâmico dos elementos voltados a prover a segurança passiva de um veículo. Conceitualmente simples, o teste consiste em se colidir um veículo contra um obstáculo dentro de condições pré-determinadas e observáveis, avaliando-se em seguida a gravidade do impacto sobre a estrutura do veículo e principalmente sobre seus ocupantes. Um aspecto importante é instrumentação do teste por meio de sensores como acelerômetros e células de impacto. Em poucos segundos, uma massa significativa de dados é coletada e armazenada por dispositivos de aquisição de dados. Esses dados são posteriormente analisados e assim se determinam os resultados do teste [Hobbs and McDonough 1998].

Todavia, uma questão que pode ser levantada diz respeito à confiabilidade dos dados coletados. Embora existam padrões que definam a sensibilidade dos sensores, frequências de operação, taxas de amostragem, protocolos de comunicação e capacidade dos dispositivos de armazenamento, não existem quaisquer requisitos relacionados à segurança cibernética dessas informações. É fato que os resultados de testes de impacto realizados sistematicamente podem afetar as relações de mercado, seja pela definição quanto à homologação de um determinado modelo de veículo ou ainda pela decisão do consumidor quanto a optar por um veículo indicado mais seguro. Sendo assim, é necessário se garantir que esses resultados derivam de informações confiáveis.

Neste trabalho, os autores apresentam uma discussão voltada à segurança cibernética dos processos de instrumentação e aquisição de dados em um teste de impacto. O termo sensoriamento seguro é utilizado como ponto de partida para se avaliar trabalhos relacionados a essa problemática. Em seguida, um modelo de ataque é descrito para se evidenciar os principais ataques que podem ser lançados contra os sistemas de sensoriamento e aquisição de dados durante a realização de um teste de impacto. Desses ataques, a maioria pode ser tratada de forma satisfatória por meio de metodologias de segurança da informação já consolidadas na literatura. Todavia, um ataque específico está relacionado com dificuldade de se identificar os sensores embarcados no veiculo durante o teste. Para tanto, os autores apresentam uma proposta original de propagação de um identificador único para cada sensor por meio de um dispositivo luminoso. A recuperação e verificação deste identificador são feitas por meio das câmeras de alta velocidade já utilizadas no teste. Com isso, pode-se além de se autenticar cada sensor utilizado, garantir que o mesmo encontra-se de fato embarcado no veículo, sem a necessidade de se acrescentar novos equipamentos ao contexto de testes.

2. Trabalhos relacionados

O conceito de sensoriamento seguro é encontrado em diversos trabalhos recentes na literatura [Sorber et al. 2012, Colak et al. 2012, Han et al. 2013]. Em muitos sistemas de controle ou de coleta de informações, o uso de sensores é cada vez mais comum. Em geral, sensores são usados para prover informação de forma automática a um sistema de tomada de decisão. A resposta do sistema como um todo depende da precisão e integridade dos dados obtidos. Entretanto, no processo de aquisição de dados, muitas vezes a informação é transmitida por diferentes canais e protocolos ou processada por elementos computacionais que podem não prover mecanismos de segurança da informação. Ao mesmo tempo, estes elementos intermediários estão sujeitos tanto a falhas quanto a ataques maliciosos, externos ou internos ao sistema. Assim, é necessário que o sistema de tomada de decisão disponha de mecanismos para avaliar as informações providas e confirmar se as mesmas são confiáveis.

Um caso interessante de sensoriamento seguro é dado por Sorber et al. (2012). Os autores consideram o sistema mHealth, que é um caso particular de PHMS (*Pervasive Health Monitoring System*). Neste sistema, sensores corporais coletam dados sobre as condições de saúde de pacientes e as transmitem a um centro médico de monitoramento. No mHealth, os sensores e os servidores que recebem os dados são considerados seguros; entretanto, o meio de transmissão, que consiste de um telefone celular, não o é, implicando que os dados dos sensores podem ser interceptados, usados indevidamente ou mesmo adulterados, resultando na emissão de diagnósticos equivocados. Os autores apresentam então uma estratégia para proteger as informações coletadas durante todo o trajeto não seguro, usando um *smart card* para autenticação dos sensores e armazenamento das informações, até que seja possível transmitir os dados pelo telefone celular. Durante a transmissão, o *smart card* é usado novamente para criptografar os dados e protegê-los durante todo o trajeto até o servidor seguro.

Um exemplo simples do uso de sensoriamento seguro na área veicular pode ser encontrado no cronotacógrafo digital europeu [Colak et al. 2012]. Tal como no Brasil, o cronotacógrafo é utilizado na Europa para controlar a jornada de trabalho de motoristas. Neste dispositivo existe um sensor de movimento não intrusivo capaz de contar o número de giros do eixo do veículo e assim determinar a distância percorrida. O sensor é ligado por meio de um cabo ao cronotacógrafo. Na primeira geração destes dispositivos, eram comuns fraudes envolvendo a substituição do sensor de movimento por um sensor malicioso. Na segunda geração de cronotacógrafos, foi introduzido o uso de sensores inteligentes, onde o sinal analógico obtido é digitalizado pela própria eletrônica embarcada. Cada sensor possui um identificador único, que é gravado apenas uma vez, em tempo de fabricação. O sensor é protegido contra adulteração (*tamper proofing*) de modo que qualquer tentativa de leitura ou modificação de seu identificador seja facilmente detectada. A transmissão dos dados para o cronotacógrafo é feita por meio de um canal autenticado e criptografado, estabelecido a partir do identificador.

Outro trabalho com ideias relacionadas é apresentado em Han et al. (2013). Neste trabalho, os autores consideram o uso do sistema Ford OpenXC, que permite ao usuário obter informações de sensores e subsistemas de seu veículo. Um telefone inteligente é utilizado para coletar informações da rede de controle veicular, processá-las e exibi-las ao usuário. O problema é que a rede de controle não faz distinção entre

um nó conectado apenas para obter informações e outro que possa também transmitir informações. Um software malicioso pode fazer uso desse canal de comunicação para envio de mensagens espúrias, comprometendo o desempenho do veículo e pondo em risco a vida de seus ocupantes. Neste trabalho os autores definem um modelo de integração segura, apresentando os cenários de ataque e definindo requisitos de segurança. Por fim, é proposto um mecanismo de verificação em três etapas, baseado em um *gateway* de segurança e processos de autenticação mútua entre dispositivos do usuário e as unidades de controle veicular.

Até onde é de conhecimento dos autores, não existem até o momento trabalhos abordando questões relacionadas ao sensoriamento seguro em testes de impacto. Ideias abstraídas dos trabalhos descritos nessa seção podem auxiliar tanto na identificação de vulnerabilidades no sensoriamento desses testes como na concepção de soluções para aumentar a segurança cibernética dos mesmos. Estas considerações serão desenvolvidas nas seções subsequentes deste trabalho.

3. Dinâmica e Sensoriamento de Testes de Impacto

3.1. Aspectos gerais de um teste de impacto

Existem tipos variados de testes de impacto, cada qual em função de finalidades, que vão desde a homologação de um veículo por fins de legislação até a pesquisa ou desenvolvimento de novos materiais e dispositivos de segurança. Para cada tipo diferente de teste, existe um protocolo que descreve a configuração física durante o impacto, as condições ambientais que devem ser observadas e também os elementos de instrumentação utilizados para coleta das informações [Hobbs and MacDonough 1998].

Durante um teste de impacto, o veículo é acelerado lentamente por meio de um mecanismo auxiliar baseado em cabos para evitar que uma aceleração brusca modifique o ajuste físico dos sensores. Além disso, é necessário que no momento da colisão a aceleração do veículo seja praticamente zero e sua velocidade constante. Próximo à área de impacto, o mecanismo de aceleração libera o veículo para que o mesmo continue sua trajetória de forma livre. A colisão do veículo ocorre sob condições de controle bem definidas, em conformidade com o protocolo de testes adotado. Durante a colisão, as dinâmicas de desaceleração, absorção de impacto e deformação dos elementos envolvidos proveem os dados necessários para a análise e resultados do teste. Essas informações serão coletadas pelo sistema de instrumentação, que é descrito a seguir.

3.2. Instrumentação de um teste de impacto

O primeiro trabalho sobre sensoriamento e instrumentação de testes de impacto é de Snider (1964). Embora as tecnologias envolvidas tenham evoluído significativamente, a metodologia usada atualmente difere muito pouco dessa proposta inicial. Nos testes de impacto modernos, a instrumentação também é definida nos protocolos de teste. Usualmente um protocolo de testes estabelece sua configuração de instrumentação levando em conta três elementos de teste principais:

1) O ATD (*Anthropomorphic Test Dummy*), um boneco humanoide usado para avaliar as consequências do impacto sobre pessoas dentro do veículo;

2) O próprio veículo, cuja instrumentação coleta informações sobre a dinâmica da colisão e os efeitos do impacto sobre a estrutura mecânica do mesmo;

3) A barreira de colisão, que também é instrumentada para se avaliar principalmente as forças de impacto envolvidas na colisão.

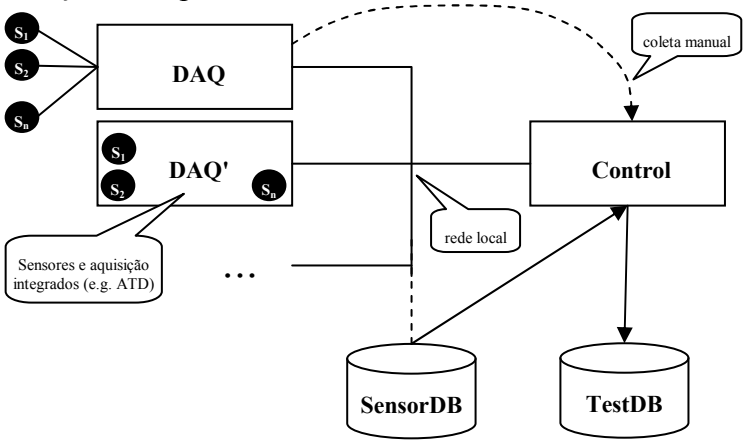


Figura 1: Visão geral do sensoriamento de um teste de impacto

A Figura 1 ilustra uma arquitetura de instrumentação geralmente adotada em testes de impacto. Ela relaciona os sensores, os dispositivos de aquisição de dados (DAQ), um banco de dados de informações de sensores (SensorDB), uma central de controle (Control) e um banco de dados de resultados dos testes (TestDB).

Os sensores são a parte crucial da instrumentação. Cada sensor possui um cadastro no banco de dados de sensores (SensorDB) que armazena informações sobre a identificação do sensor, propriedades físicas e parâmetros de calibração. O banco de dados é alimentado e revisado durante o processo de calibração do sensor, que ocorre sempre antes da realização de um teste de impacto.

Duas configurações são comuns na comunicação entre os sensores e o DAQ. Na primeira os sensores são acoplados ao DAQ durante a preparação do teste. Usualmente os sensores são analógicos e o DAQ funciona como conversor analógico/digital e armazenador temporário das informações. Em alguns casos podem-se utilizar sensores inteligentes, os quais executam uma parte do processamento do sinal em sua própria eletrônica. Outra configuração possível é quando sensores e DAQ são integrados pelo fabricante, de maneira que a unidade de instrumentação possa ser vista como um único bloco funcional. O ATD é um exemplo deste caso de instrumentação, onde sensores e DAQ formam um único conjunto. Nas duas configurações, os sistemas DAQ são elementos para controle, armazenamento temporário e disponibilização dos dados. No aspecto de controle, o DAQ é muitas vezes capaz de identificar e testar cada sensor a ele conectado e verificar se este corresponde a uma configuração esperada. O DAQ também contém memórias internas para armazenamento das informações coletadas. Alguns modelos permitem a transferência dos dados por meio de remoção da memória do dispositivo, conforme ilustrado.

O centro de controle (Control) tem a função de controlar as atividades de verificação da instrumentação e aquisição dos dados de teste. O *software* em Control pode inclusive fazer verificação de consistência dos dados, analisando a resposta de um

sensor em relação aos demais. Concluído o teste, a análise dos dados também será feita por Control, para posterior disponibilização dos resultados em TestDB.

Um elemento que não é parte direta da instrumentação, mas que é absolutamente relevante neste estudo é o sistema de câmeras de alta velocidade. O uso do vídeo para inferir informações sobre os resultados de um teste de impacto é algo bem consolidado. Sistemas comerciais de aquisição e processamento de imagens usualmente permitem a identificação de elementos físicos no vídeo. O vídeo pode inclusive fornecer informações de sensoriamento, como determinar a aceleração do veículo ou de um ATD interno a ele em função da triangulação de três pontos conhecidos na imagem.

4. Modelo de Ataque

Nesta seção é proposto um modelo de ataque visando identificar as principais ameaças e vulnerabilidades associadas ao sensoriamento e aquisição de dados de um teste de impacto. Um ataque pode ser motivado por diversas razões, conforme descrito na introdução deste trabalho. Os resultados dos testes podem afetar significativamente as decisões de mercado quanto à aquisição de um veículo ou não. Atores internos e externos ao teste, incluindo pessoal técnico, podem sofrer pressão ou mesmo suborno para interferirem nos resultados dos testes de modo a favorecer interesses específicos.

Como ponto de partida, são propostas três condições iniciais a serem satisfeitas para se garantir que o sensoriamento de um teste de impacto ocorre de forma confiável:

1) Todos os sensores utilizados no teste foram previamente calibrados.
2) Os sensores foram autenticados e suas medições estão integras.
3) Os sensores foram de fato utilizados no teste de impacto respectivo.

Cada uma das condições citadas é discutida em pormenores nas seções subsequentes, bem como os ataques que podem ocorrer em violação a essas condições.

4.1. Calibração dos sensores

Conforme descrito na seção 3, existe um processo prévio à realização dos testes que é a calibração dos sensores. A calibração é feita em laboratório. Os dados para correção das medições obtidas de cada sensor são inseridos em SensorDB e a partir dele disponibilizados para Control, e eventualmente ao próprio DAQ.

Com relação a esta condição, os seguintes ataques são identificados:

A1 - Modificação deliberada ou acidental dos dados de calibração dos sensores. Este ataque é lançado contra SensorDB e pode ocorrer antes do teste, no momento da calibração de um sensor ou ainda em posteriormente, por ação deliberada do atacante.

A2 - Negação de Serviço (DoS) em SensorDB. Este ataque consiste em tornar SensorDB indisponível para acesso, seja pelo envio excessivo de pacotes de requisição ao banco de dados ou ainda por quebra do canal de comunicação.

Em uma breve análise, ambos os ataques exploram vulnerabilidades cujas alternativas de solução são bem conhecidas. Entende-se que a segurança de um sistema de banco de dados é um assunto extensamente abordado na literatura, e diversas abordagens são passíveis de aplicação aqui.

4.2. Autenticação dos sensores e integridade das medições

Esta condição está associada especificamente aos sensores, ao DAQ e às interações entre eles que ocorrem durante o teste de impacto. Qualquer erro ou falha pode implicar na perda de dados e consequentemente na invalidação do teste.

Com relação a esta condição, os seguintes ataques são identificados:

A3 - Substituição de sensor calibrado por um não calibrado. Um atacante que tenha acesso ao arranjo físico dos sensores pode substituir o sensor esperado por um devidamente modificado para gerar medições espúrias.

A4 - Perturbação dos parâmetros físicos do sensor. Tal como em *A3*, um atacante pode inserir qualquer elemento que perturbe aspectos físicos do sensoriamento.

A5 - Envio de comandos espúrios para o DAQ. Quando o DAQ suporta comandos remotos, um atacante pode explorar a segurança destas interfaces.

A6 - Modificação deliberada ou acidental dos dados coletados. Este ataque pode incidir de diferentes formas, em quaisquer dos canais de comunicação disponibilizados pelos sensores e pelo DAQ, ou ainda nos dados armazenados pelo DAQ.

Dos ataques mencionados, *A3* e *A4* são ataques de natureza física. A substituição de um sensor calibrado por outro malicioso pode ser prevenida por autenticação do sensor. A possibilidade de se ter mecanismos para tal depende da tecnologia utilizada. Sensores analógicos, por exemplo, não possuem eletrônica suficiente para implementar uma autenticação tradicional. Neste caso, o mais adequado é que o sensor e o componente de digitalização do DAQ sejam vistos como um único elemento no escopo da autenticação. Já para os sensores inteligentes podem ser propostos mecanismos de autenticação mais sofisticados. De fato, algumas soluções comerciais de sensoriamento de testes de impacto baseadas em sensores inteligentes alegam possuir mecanismos de autenticação. Por serem soluções fechadas não é possível avaliar sua eficácia. Entretanto, existem soluções para esse problema na literatura, com é o caso do cronotacógrafo digital citado na seção 2. Por sua vez, a perturbação de parâmetros físicos do sensor pode ser detectada por algoritmos que avaliem sua resposta. Uma perturbação não será capaz de produzir respostas coerentes, e muito provavelmente afetará outros sensores, podendo ser identificada por sistemas de detecção de anomalias.

O ataque *A5* constitui também um problema de autenticação. Uma vez que o DAQ passe a exigir credenciais para qualquer execução de comandos em suas interfaces, esse ataque é mitigado e mesmo eliminado se o sistema de autenticação for suficientemente seguro. Estas solução já existem para muitos dispositivos e o poder computacional disponível no DAQ é suficiente para implementar uma delas.

O ataque *A6*, por sua vez, diz respeito especificamente à garantia de integridade das informações. Este ataque pode ocorrer em diferentes partes no fluxo de informações do processo de teste, uma vez que a informação pode ser corrompida já no momento de sua digitalização pelo DAQ, ou ainda no envio da mesma para Control. Independente do momento quando ocorre o ataque, diferentes técnicas para prover integridade podem ser utilizadas. É possível, por exemplo, se prevenir ataques de falsificação da informação protegendo-se os canais de troca de dados, por meio do uso de tecnologias vastamente exploradas. É possível ainda o uso de mecanismos para verificação da integridade da

informação em cada etapa, rastreando sua origem ao ponto mais primário possível. No caso dos sensores inteligentes, é possível estabelecer uma política de chave pública, sendo que cada sensor tem uma chave privada e suas medições são assinadas digitalmente. Embora abordagens similares possam apresentar complexidade em termos de custo computacional e implementação, existem propostas eficientes para tal.

4.3. Verificação da utilização dos sensores no teste de impacto

Embora em um primeiro momento esta condição pareça trivial, sua verificação é importante em face de dois ataques relativamente simples, todavia plausíveis. Estes ataques são descritos a seguir:

A7 - Uso de sensores diferentes daqueles que foram previamente calibrados. Essa é uma variação do ataque descrito em *A3*, todavia se considera aqui que um arranjo completo de sensores foi modificado, como a substituição de um ATD completo.

A8 – Uso dos sensores calibrados em um ambiente que diferente do teste de impacto. Este ataque constitui uma tentativa deliberada de fraude, pois envolve a preparação de um ambiente específico para se produzir dados de teste falsos.

No ataque *A7*, a substituição dos sensores pode ocorrer por razões não intencionais ou mesmo intencionais. Em uma situação não intencional, por exemplo, pode haver por parte dos responsáveis a intenção de omitir o ocorrido, em virtude dos custos envolvidos em uma eventual repetição dos testes. Entretanto, a solução para o problema é a mesma descrita em *A3*.

O ataque *A8*, por sua vez, difere dos demais ataques pelo fato de que a autenticação dos sensores e aquisição de dados pode ocorrer normalmente, dentro do ambiente forjado para os testes. Considere-se como ilustração uma situação onde um veículo será submetido a um teste de impacto de homologação. Em paralelo, um ambiente a parte é preparado para simular uma colisão cujo comportamento dinâmico estará em conformidade com parâmetros avaliados. Ao mesmo tempo em que o veículo é submetido ao teste, o conjunto de instrumentação que deveria estar embarcado no veículo é "testado" no ambiente de simulação. A coleta de dados ocorre normalmente, estes são transmitidos e disponibilizados para a análise em Control. Os sensores foram autenticados, os dados estão íntegros, mas os sensores não estavam dentro de veículo.

Conforme discutido nesta seção, todos os ataques identificados, com exceção de *A8*, possuem contramedidas conhecidas na literatura, algumas delas muito bem consolidadas, tais como os mecanismos baseados em assinatura digital para prover integridade das informações. No entanto, no caso do ataque *A8*, não é do conhecimento dos autores abordagens que se proponham a solucionar o problema. Na próxima seção deste trabalho, é apresentada uma ideia original para o mesmo, que se baseia no uso dos sistemas de visão computacional para identificação física dos sensores utilizados.

5. Proposta para autenticação dos sensores a partir de imagens

Na seção anterior, foi apresentado que o ataque *A8* explora condições específicas de fraude na qual um teste de impacto ocorre sem que sua instrumentação esteja de fato embarcada no veículo. Para se evitar este ataque é necessário o uso de algum mecanismo que permita identificar os sensores embarcados dentro do veículo e

confirmar que os mesmos correspondem àqueles previstos para tal na etapa de calibração. Intuitivamente, é natural que se considere o uso de tecnologias comuns para transmissão de um identificador único, como RFID ou redes sem fio. Entretanto, conforme discutido na seção anterior, tecnologias que se baseiam unicamente na transmissão de dados podem ser utilizadas no conjunto correto de sensores em um teste simulado, que ocorre em paralelo com o teste real. Outras tecnologias, por sua vez, podem requerer a instalação de antenas e equipamentos adicionais, poluindo fisicamente o espaço destinado à colisão do veículo e mesmo interferindo nos resultados dos testes.

Como alternativa nossa proposta se baseia no uso das câmeras de alta velocidade, já presentes no cenário de teste, para identificar os sensores utilizados. A identificação é feita por meio de um sinal luminoso processado no próprio vídeo, que permite identificar de forma única um determinado sensor dentro do campo visual da câmera.

Para viabilizar a proposta, assumem-se as seguintes pré-condições:

a) Cada sensor a ser identificado possui associado a ele um dispositivo luminoso visível (por exemplo, um LED), seja sobre a superfície do veículo ou interno a este. O dispositivo luminoso é acionado pelo próprio sensor ou então pelo DAQ respectivo, caso se trate de um sensor analógico;

b) Cada sensor a ser identificado está dentro do campo visual de alguma das câmeras usadas no teste de impacto, dentro do período de intervalo definido para que o sensor emita seu identificador visual.

O período de intervalo em questão corresponde ao tempo disponível durante o teste de impacto no qual cada sensor deve emitir sua identificação, por meio de seu dispositivo luminoso. Essa identificação corresponde a um código de autenticação exclusivo do sensor, que pode ser verificado por meio de uma política de chaves. O identificador é emitido usando-se uma representação binária no dispositivo luminoso (zero quando apagado, um quando aceso) e esta informação pode ser recuperada por meio de processamento do vídeo associado à câmera respectiva. Cada um desses aspectos será tratado em detalhes nas subseções que se seguem.

5.1. Tempo disponível para propagação do identificador de um sensor

Um aspecto prático que precisa ser considerado é a questão do tempo disponível para propagação do identificador de cada sensor por meio do dispositivo luminoso. Para tanto, é necessário considerar não apenas o tempo real disponível, como também a disponibilidade de recursos para processamento desta informação.

Durante o teste de impacto, após o veículo ser liberado pelo mecanismo auxiliar de aceleração, o tempo que envolve a colisão é muito curto. Segundo Snider (1967), todos os eventos relevantes (desaceleração, a absorção do impacto por elementos de segurança e deformação do veículo) ocorrem em um tempo aproximado de 200 milissegundos. Todavia, se for considerado o tempo no qual o veículo está próximo à área de impacto, ou seja, desde sua liberação até a parada após a colisão, pode-se estabelecer um período de tempo entre 1 a 2 segundos. Por questões práticas, estamos considerando o tempo de um segundo como o disponível para que os sensores emitam seus respectivos identificadores. A proximidade do veículo com a área de impacto é

essencial porque nesta condição existem mais câmeras disponíveis para monitoramento de diferentes grupos de sensores.

5.2. Processamento do identificador exibido por dispositivo luminoso

Como já apresentado em linhas gerais, este trabalho propõe que o identificador exibido pelo sensor por meio do dispositivo luminoso seja recuperado a partir do processamento de vídeo de uma câmera de alta velocidade utilizada no teste de impacto. Tal como descrito na seção 3, o processamento de vídeo para se inferir resultados de um teste de impacto é algo usual nas ferramentas comerciais atualmente disponíveis. Assim, o que se propõe aqui é que os mesmos recursos sejam utilizados para se processar um sinal que é propagado em vídeo por meio do dispositivo luminoso do sensor.

Para tanto, considere-se o vídeo gerado por uma câmera durante o teste de impacto, e que o mesmo possui em sua área de imagem um determinado sensor para os qual se deseja recuperar o identificador emitido. Espera-se que a cada quadro seja possível identificar o sinal luminoso por meio de processamento de imagem, determinando-se se o mesmo encontra-se apagado ou aceso, o que em codificação binária equivale à representação de zero ou um respectivamente. Os principais protocolos de definição de testes de impacto estabelecem que a taxa mínima de quadros gerados por uma câmera deve ser de mil quadros por segundo, que para fins de praticidade consideramos com 1024 quadros. Em circunstâncias ideais, seria possível recuperar neste intervalo de tempo 1024 bits, fazendo com que os dispositivos luminosos sejam sincronizados com o sinal de vídeo da câmera. No entanto, a instalação de uma estrutura para sincronismo de sinal no ambiente de testes pode ser complexa. Seria necessário prever o uso de diversos cabos de sincronismo ligados à câmera ou ainda um sistema de sincronismo sem fio, o que pode não ser factível. Será analisada assim a possibilidade de propagação e captura do identificador de forma assíncrona.

Considere-se inicialmente que os sensores são programados para propagar seu identificador na mesma frequência de captura de quadros da câmera. No entanto, pelo fato de não haver um sincronismo de sinal, algumas condições especificas podem ocorrer, as quais são descritas a seguir.

Primeiramente, em virtude do erro acumulado na diferença de frequência, em algum momento pode ocorrer uma das seguintes situações: a perda de um bit quando a frequência de propagação do identificador é levemente maior do que a frequência interna da câmera ou a leitura duplicada de um bit quando o oposto. Esse problema é um caso comum na transmissão de dados digitais sobre um canal não confiável, e pode ser tratada pelo uso de um algoritmo de FEC (*Forward Error Correction*), levando-se em consideração apenas o *overhead* de bits de correção de dados.

A segunda condição é que, em função da defasagem do sinal, existe a possibilidade remota do dispositivo luminoso ter sua transição de sinal sobreposta ao instante de abertura do obturador da câmera. Em consequência, este sinal assumiria no frame de vídeo um estado indeterminado, sem que o algoritmo de processamento de imagens consiga determinar com exatidão se seu valor é zero ou um. Este problema pode ser eliminado se a frequência de propagação do identificador for definida como a metade da frequência de captura de quadros da câmera, de modo que é esperado que

cada bit do identificador seja propagado em dois frames. Em consequência, a disponibilidade de bits para composição do identificador é reduzida pela metade.

Em face destas condições, nossa proposta define que o identificador deve ser propagado com a metade da frequência disponível para captura de vídeo, o que equivale a 500 Hz, o que arredondamos para fins de praticidade para 512 Hz. Esta alternativa tem ainda como vantagem o fato de tornar desnecessário o uso de um algoritmo de FEC, uma vez que a perda ou duplicação de bits pode ser apropriadamente tratada com a duplicação dos mesmos. Deste modo, nossa proposta passa a considerar que se dispõe de um pacote de até 512 bits para encapsulamento do identificador de um sensor.

5.3. Composição do identificador de um sensor

Para a composição do identificador do sensor, alguns aspectos precisam ser levados em consideração. Primeiramente, esse identificador deve ser único e sua verificação deve estar fortemente associada ao sensor, de modo a garantir que somente este seja capaz de gerar o identificador. Um aspecto secundário diz respeito à propagação deste sinal. Por se tratar de um processamento assíncrono, não é possível transmitir apenas o identificador; é necessário se acrescentar um preâmbulo ao pacote, que sirva como sinal de sincronismo, viabilizando assim o processamento do mesmo pelo sistema de visão computacional. Por fim, preâmbulo de sincronismo e identificador devem ser acomodados dentro de um pacote limite de 512 bits, conforme visto na seção 5.2.

Para garantir que o identificador seja único, nossa proposta se baseia no uso de uma política de chave pública. Para tanto consideramos que cada sensor possui um par de chaves. A chave privada é embarcada no sensor, e não pode ser extraída deste. A chave pública é informada pelo sensor no momento da calibração e faz parte dos dados disponíveis em SensorDB. No caso dos sensores analógicos, é possível propor que a atribuição das chaves estaria associada ao canal do DAQ respectivo, observando-se as mesmas premissas já definidas quanto à política de chaves.

Uma vez definido o uso de um par de chaves, um identificador único pode ser obtido por meio da assinatura digital de uma informação conhecida tanto pelo sensor quanto pelo processo responsável pela verificação deste identificador. Por praticidade, definiremos esta informação como R. O valor de R pode ser definido de diferentes formas. Um exemplo é a concatenação dos últimos n valores aferidos pelo sensor imediatamente antes do início da propagação do identificador pelo dispositivo luminoso, ou ainda simplesmente um valor aleatório qualquer. Deste modo, para se obter o identificador I de um sensor qualquer, define-se a seguinte expressão:

$$I = R \oplus sign(K, hash(R))$$

Onde *hash()* e *sign()* são respectivamente funções para geração de um resumo criptográfico e para criptografar um *string* usando a chave privada K do sensor, e $\oplus$ o operador de concatenação.

Resta agora definir quais algoritmos criptográficos devem ser adotados para as funções *hash()* e *sign()* em função do tamanho L definido como o número máximo de bits disponíveis para propagação do identificador. Seja P o preambulo de dados usado como sinal de sincronismo na propagação do identificador, tem-se que:

$$L \geq length(P \oplus I)$$

Onde *length()* é a função que informa o comprimento de um *string* de texto.

Pelas conjecturas definidas na seção 5.2., temos que o valor de L deve ser inferior a 512 bits. Portanto, o mesmo se aplica ao comprimento do preambulo de sincronismo concatenado com o identificador. O identificador, por sua vez, tem como prefixo o valor de R. A princípio, ambos P e R podem ter seus tamanhos arbitrados. Considerando-se que 16 bits sejam suficientes para cada um deles, temos um restante de 480 bits disponível para a assinatura digital de R, que é efetivamente a parte mais importante do identificador. Este tamanho impede o uso do algoritmo RSA para tal aplicação, uma vez que o tamanho mínimo para uma chave RSA, conforme recomendado pelo NIST, é de 1024 bits. A alternativa é, portanto, o uso de um algoritmo de curvas elípticas, que propicia o mesmo nível de segurança com chaves bem menores. Além disso, os algoritmos de curvas elípticas são mais adequados para implementação em dispositivos com recursos computacionais limitados, como é o caso de um sensor.

Em face disso, optamos por adotar os mesmos algoritmos usados em Camara et al. (2012), que são a função SHA-224 como *hash()* e algoritmo ECDSA 224 bits como *sign()*, equivalente em segurança ao RSA com chaves de 2048 bits. As funções escolhidas resultam em uma assinatura digital de 448 bits, suficiente para o tamanho limite de 480 bits já determinado em função de P e R. Ao mesmo tempo em que satisfazem as condições estabelecidas para propagação do identificador, estes algoritmos permitem a geração de um identificador único, com um elevado grau de confiabilidade.

6. Experimentos e discussões

6.1. Experimento prático com uma câmera de alta velocidade

Com o objetivo de demonstrar a factibilidade da proposta apresentada, foi realizado um experimento com uma câmera de alta velocidade para se demonstrar que é possível recuperar um identificador propagado por um sinal luminoso a partir do processamento de vídeo.

O experimento foi realizado utilizando um LED conectado à saída de um gerador de sinais, modelo 33220A do fabricante Agilent, para simular o envio de dados do sensor na frequência de 500 Hz. O sinal gerado constitui em uma onda quadrada com *duty cycle* de 50 % e amplitude variando de 0 a 5 Volts. Uma câmera de alta velocidade, modelo M310 do fabricante Vision Research, foi configurada com de resolução de 1200x800 e taxa de aquisição de 1000 quadros por segundo para realizar um teste de validação dos dados simulados pelo LED. A Figura 2 mostra o experimento monitorado a partir de um osciloscópio. A linha amarela indica o sinal de captura de vídeo, enquanto o sinal verde mostra a onda quadrada usada para ativação do LED. Como é possível observar, o momento de captura do quadro, que ocorre na borda de descida do sinal, coincide com posições nas quais o sinal do LED representa estados diferentes, que para a onda quadrada em questão correspondem aos valores zero e um. No exemplo demonstrado, cada quadro captura um bit diferente de informação, evidenciando que possível se recuperar um identificador propagado no sinal de vídeo.

Figura 2: Experimento prático demonstrando recuperação do identificador

6.2. Discussões e próximos passos

O resultado do experimento realizado teve por objetivo demonstrar a viabilidade da ideia proposta para recuperação do identificador de determinado sensor usando o próprio sistema de câmeras já presente em um teste de impacto. Entretanto, diversos aspectos práticos da implementação dessa proposta podem ser citados e endereçados como os próximos passos em nossa investigação.

O primeiro aspecto diz respeito à própria implementação do firmware responsável pelo cálculo e propagação do identificador. Embora os sensores inteligentes modernos tenham avançado significativamente em termos de recursos computacionais, é necessário se estimar quais os requisitos mínimos necessários para que este dispositivo suporte as funcionalidades requeridas pelo firmware em questão. Uma alternativa já cogitada é que o identificador seja gerado pelo próprio DAQ, o qual pode ser facilmente dimensionado para suprir esses recursos computacionais.

Outro aspecto está relacionado ao processamento de vídeo para recuperação do identificador. A literatura apresenta diversos algoritmos de visão computacional voltados para a identificação e acompanhamento (*tracking*) de elementos de interesse [Yilmaz et al. 2006]. Entretanto, em um teste impacto, os sensores sofrerão movimentos bruscos em função da propagação da onda de choque do veículo com a barreira. Em função deste movimento indeterminado, algumas dificuldades podem se apresentar, como a obstrução do sensor no campo visual da câmera por um determinado instante, por exemplo. Consequentemente, este aspecto do problema requer uma investigação mais profunda, para se identificar quais algoritmos melhor se adequam às condições descritas. Uma alternativa que pode ser investigada é o uso de mais de uma câmera por sensor, de modo que informações que venham a ser perdidas por uma câmera possam ser recuperadas a partir do vídeo de câmeras secundárias.

7. Conclusão

Neste artigo foi apresentado um estudo amplo sobre a segurança cibernética no sensoriamento e instrumentação dos testes de impacto de veículos. Como apresentado, estes testes são cruciais para se garantir que um determinado veículo atende a requisitos

específicos de segurança. A confiabilidade das informações de teste é, por sua vez, requisito para se legitimar os resultados do mesmo.

São duas as principais contribuições deste trabalho. A primeira é a apresentação do modelo de ataque descrevendo em detalhes ameaças e vulnerabilidades às quais um teste de impacto está sujeito, e que podem ser exploradas de diferentes formas por um atacante mal intencionado. A segunda é a ideia original de atribuir a cada sensor um identificador único que é propagado por um dispositivo luminoso durante o teste de impacto e pode ser recuperado usando-se o sistema de câmeras de alta velocidade já utilizado nos testes. Essa ideia pode ser explorada de diversas formas, e por si só abre espaço para investigações ainda mais detalhadas no campo de visão computacional e tratamento de sinais digitais.

Referências

Bacchieri, G. and Barros, A. J. D. (2011) "Acidentes de trânsito no Brasil de 1998 a 2010: muitas mudanças e poucos resultados", Saúde Pública, 45(5), p. 949–963.

Camara, S., Machado, R., Pirmez, L. and Carmo, L. F. R. C. (2012), "Uma arquitetura de segurança para medidores inteligentes – verificação prática de dados de energia multitarifada", Anais do XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), p. 221-234.

Caveney, D. (2010). "Cooperative Vehicular Safety Applications. IEEE Control Systems", 30(4), p. 38–53.

Colak, M., Bishop, J., Nordvik, P. J., Mahieu, V. and Loeschner, J. (2012), "Cryptographic security mechanisms of the next generation digital tachograph system and future considerations". In: Joint Research Centre Scientific and Policy Report, European Commission, Ispra, Italy.

Han, K., Potluri, S. D. and Shin, K. (2013), "On authentication in a connected vehicle: secure integration of mobile devices with vehicular networks", In: ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2013, p. 160–169.

Hobbs, C. A. and McDonough, P. J. (1998). "Development of the European new car assessment programme (Euro NCAP)", Regulation, 44, p. 3.

Paine, M. and Haley, J. (2008). "Crash testing for safety - possible enhancements to ANCAP test and rating methods", In: Australasian Road Safety Research Policing Education Conference, p. 33–42.

Snider, H. P. (1964), "Vehicle Instrumentation for Crash Testing", In: *IEEE Transactions on Industrial Electronics and Control Instrumentation*, 11(1), p. 44–49.

Sorber, J., Shin, M., Peterson, R. and Kotz, D. (2012), "Plug-n-Trust: Practical Trusted Sensing for mHealth", In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, p. 309–322.

Yilmaz, A., Javed, O. and Shah, M. (2006), "Object Tracking: A Survey", In: ACM Journal of Computing Surveys, 38(4), p. 1-45.

Attacks on single-pass confidentiality modes of operation

Olivier Markowitch , Jorge Nakahara Jr*

[1]Departement d'Informatique, Université Libre de Bruxelles, Brussels, Belgium,

{olivier.markowitch, jorge.nakahara}@ulb.ac.be

Abstract. *The main contributions of this paper are efficient distinguishing attacks against block ciphers that are conventionally modeled as pseudorandom permutations (PRP). Formally, block ciphers operate on fixed-length blocks of n bits, for example, $n = 128$ for the Advanced Encryption Standard (AES). Our analysis takes place in the setting in which the messages are m bits long, representing the entire input plaintext, where m is variable and unrelated to n. We show distinguish-from-random attacks for any n-bit block cipher in the standard modes of operation for confidentiality: ECB, CBC, CFB, OFB, CTR and XTS. We demonstrate that in all these 1-pass modes* **any** *n-bit block cipher leaves 'footprints' that allows an adversary to efficiently (in time and memory) distinguish them from a random permutation. We claim that two passes (in opposite directions) over the m-bit message, with text-dependent feedforward (chaining) and in streaming mode are sufficient to circumvent the presented attacks.*

Keywords: left-to-right diffusion, distinguishing attacks, modes of operation, (super-)pseudorandom permutations, IND-KPA, IND-CPA.

1. Introduction

Block ciphers are length-preserving cryptographic primitives that operate on finite, fixed-length text blocks. More precisely, block ciphers are keyed permutations, denoted $E_K : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, where n is a fixed integer denoting the size of one text block, and the secret key K is chosen uniformly at random from a sufficiently large key space $\mathcal{K}$. In general, n is a small integer value such as $n = 32$ for KATAN32, $n = 64$ for DES, $n = 96$ for BKSQ [Daemen and Rijmen 2000] and $n = 128$ for the AES [FIPS197 2001]. Larger values such as $n = 4096$ were adopted by the Mercy cipher [Crowley 2000]. The value of n is arbitrary and set up by convenience according to design considerations and for specific applications; n does not need to be an even integer nor a power of two. For instance, in the CTC cipher, $n = 255$ bits [Courtois 2006].

We assume the size of the key K to be large enough, say $|K| \geq 128$ bits, and subkeys to be generated efficiently and securely. Our analysis is independent of the key size or its value. Also, we do not exploit the existence (or not) of equivalent keys, weak keys [Menezes et al. 1997], complementation property (DES) or other weaknesses in the key schedule algorithm. Moreover, our attacks are in the single-key model (no related keys).

Traditionally, secure n-bit block ciphers are modeled as pseudorandom permutations (PRP) [Luby and Rackoff 1988]. It means that computationally bounded adversaries A, allowed a polynomial number q of queries, may distinguish a given block cipher E from a random permutation π, chosen uniformly at random from the set RP^n of $2^n!$ permutations, with

*Research funded by INNOVIRIS, the Brussels Institute for Research and Innovation, under the ICT Impulse program CRYPTASC.

negligible advantage

$$\text{Adv}_A(q) = |\text{Pr}(k \xleftarrow{\$} \mathcal{K} : A^{E_K} = 1) - \text{Pr}(\pi \xleftarrow{\$} \text{RP}^n : A^{\pi} = 1)|,$$

where $y \xleftarrow{\$} \mathcal{Y}$ means y is selected uniformly at random from the set $\mathcal{Y}$, and A^X returns '1' if A believes it is dealing with oracle X; otherwise, A returns '0'. Therefore, '1' means 'success' while '0' means 'failure'; 'negligible' means that the advantage grows slower than the inverse of any polynomial (in n). If the advantage is negligible even if the adversary is allowed decryption queries ($D_K = E_K^{-1}$) then, the block cipher is called a strong pseudorandom permutation (SPRP).

For negligible advantage, if the (encryption) queries are only **known** by the adversary, then the block cipher is deemed indistinguishable under known-plaintext attacks (IND-KPA); if the (encryption) queries are **chosen** by the adversary, then the block cipher is deemed indistinguishable under a chosen-plaintext attack (IND-CPA); if the decryption queries are **chosen non-adaptively** by the adversary, then the block cipher is deemed indistinguishable under a (non-adaptive) chosen-ciphertext attack (IND-CCA1); if the decryption queries are adaptively chosen by the adversary, then the block cipher is deemed indistinguishable under an adaptively chosen-ciphertext attack (IND-CCA2). If the advantage is non negligible for a single adversary, then if the queries are known to the adversary, the block cipher is not IND-KPA. Analogously, for chosen queries the block cipher is not IND-CPA, and so on.

In practice, real messages are m bits long, with m variable and unrelated to n. A naive solution to provide confidentiality in all cases would be to have block ciphers defined for every possible value of m, but this is not realistic. Rather, modes of operation [Dworkin 2001, IEEE 2008] are defined to extend the domain of application of E_K from $\mathbb{Z}_2^n$ (one text block) to $\mathbb{Z}_2^m$ (the full message), where m may be arbitrarily large but is always finite. Informally, a secure mode of operation should not disclose the fact that it is using an n-bit block cipher E_K as a building block. In summary, a secure mode should turn an n-bit (S)PRP into an m-bit (S)PRP. Consequently, issues such as padding, ciphertext expansion, blockwise or bitwise diffusion, unidirectional diffusion should be avoided, that is, weaknesses in the underlying n-bit block cipher should not propagate to the larger m-bit block.

Standard confidentiality modes of operation nowadays include: Electronic CodeBook (ECB), Cipher Block Chaining (CBC), Output FeedBack (OFB), Cipher FeedBack (CFB), Counter (CTR) and XEX Tweakable block cipher with ciphertext Stealing (XTS) [Dworkin 2001, Dworkin 2010a, IEEE 2008, Rogaway 2004]. The XTS mode in [IEEE 2008] was explicitly instantiated with the AES as the underlying block cipher. Moreover, the maximum size m of a message allowed to be encrypted via XTS-AES was upperbounded to 2^{20} 128-bit blocks. The fact that these modes of operation are linked to the AES [Dworkin 2001, Dworkin 2010a] makes this analysis quite relevant nowadays[1].

We consider random permutations operating directly on m-bit strings, and not n-bitwise like E_K, whether n is even, odd, a power of two, a divisor of m or otherwise. Moreover, random permutations are not structured transformations that require modes of operation or Feistel or SPN structures like E_K. A random permutation, in our context, is a bijective mapping chosen at random from the set of all $2^m!$ permutations of the space $\{0,1\}^m$ of m-bit strings. We

[1] FIPS standards, such as NIST SP800-38E and SP800-38A, do not include any analysis at all of any mode of operation, not even about any limitations of these modes.

denote a random permutation as $\pi \xleftarrow{\$} \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2^m$, a mapping selected at random from the set $\text{RP}^m = \{\pi_i^m : 1 \leq i \leq 2^m!\}$.

In this paper, instead of forcing random permutations to operate on n-bit strings, abiding to a block cipher domain size, we look at how block ciphers fare when forced to operate on m-bit strings for arbitrary, variable m, which is unrelated to n. In other words, instead of 'downsizing' the random permutation to always operate on fixed n-bit blocks, we work the other way around: we operate on m-bit blocks from the start because m represents the real size of an entire input message. Consequently, the queries made by an adversary are m bits long, which may be smaller, equal or larger than n bits. Before each attack starts, we set a value for m and do not change it until the attack ends.

In a block cipher setting, both an n-bit block and a full m-bit message are usually called plaintext. To make the distinction clear for our attacks, n is bound to a block cipher domain space, like $n = 64$ bits for the DES, while m is bound to a full input text message, for instance, the Project Gutenberg (ASCII) copy of the King James Bible (Old and New testaments) is 4.13 Mbytes or $m = 34,663,312$ bits long. To avoid extreme cases such as $m = O(2^n)$, we restrict our analysis to m being a polynomial in n: $m = O(n^t)$ for t a fixed constant unrelated to n. Otherwise, the adversary could cheat by using a single 2^n n-bit long message that contains all n-bit values. Further, depending on the mode of operation used, this single, long message could provide the entire codebook, which allows one to encrypt and decrypt any n-bit block without knowing the key.

This paper is organized as follows: Sect. 2 lists our contributions; Sect. 3 briefly describes the confidentiality modes under analysis; Sect. 4 describes distinguishing attacks in a PRP setting that apply to any block cipher; Sect. 5 discusses 2-pass modes and how they counter the attacks described in the previous section. Sect. 6 lists our conclusions.

2. Contributions

Our contributions address real **limitations/shortcomings of standard single-pass confidentiality modes of operation**. We explain systematically and constructively, for all these modes, how to perform efficient distinguishing attacks in a PRP setting. We describe attacks that

(i) work in a black-box setting, which in our case means the attacks work for any block cipher and any key schedule algorithm,
(ii) are very efficient concerning time, data and memory complexities, and thus violate any reasonable security thresholds whether in theory or in practice,
(iii) have very high success rate,
(iv) do not depend on (and cannot be countered by changing) the key size, key value, number of rounds, IV or nonces.

3. Brief Description of Confidentiality Modes

We briefly summarize the modes of operation under analysis in this paper. Let $P = (P_1, P_2, \ldots, P_t)$ denote an m-bit plaintext message and $C = (C_1, C_2, \ldots, C_t)$ denote the corresponding ciphertext, where $m = \sum_{i=1}^{t} |P_i|$. Some modes require random n-bit initial values, denoted IV.

- ECB: in the Electronic CodeBook mode each ciphertext block C_i depends only on P_i according to the formula $C_i = E_K(P_i)$ for $i \geq 1$. Diffusion in ECB is blockwise, which

means limited error propagation and independent (parallel) encryption of blocks. There may be a need for padding, if $m \not\equiv 0 \bmod n$.

- CBC: in Cipher Block Chaining mode, each ciphertext block is chained to the previous cipher block, and thus, depends on all previous plaintext blocks, according to the formula $C_i = E_K(P_i \oplus C_{i-1})$, for $i \geq 1$, and $C_0 =$ IV. Diffusion is better compared to the ECB mode, due to text chaining in left-to-right direction. On the other hand, parallel processing is hindered because of chaining. Error propagation is limited: if C_i is damaged, only P_i and P_{i+1} are affected.
- b-bit CFB: b-bit Cipher FeedBack mode turns a block cipher into a self-synchronous stream cipher according to the formula $C_i = P_i \oplus \text{lsb}_b(E_K(C_{i-1}))$, for $i \geq 1$ and $C_0 =$ IV, where $\text{lsb}_b(x)$ denotes the least significant b bits of x. Due to the ciphertext chaining C_{i-1}, this mode is non parallelizable. On the other hand, the CFB mode only needs E_K for both the encryption and decryption operations. Moreover, there is no need for padding, since it is a streaming mode operating on variable-length blocks.
- b-bit OFB: b-bit Output FeedBack mode turns a block cipher into a synchronous stream cipher according to the formula $C_i = P_i \oplus X_i$, where $X_1 = \text{lsb}_b(E_K(\text{IV}))$ and $X_i = E_K(X_{i-1})$ for $i > 1$. In this paper, we assume $b = n$. Note that the key stream exclusive-ored to P_i is text independent, i.e. unlike the CFB mode, X_i only depends on the IV and the key K. Therefore, OFB is a parallelizable mode since X_i can be precomputed (and stored) beforehand. It also means independent blocks P_i can be (re)encrypted without affecting C_j whether $j < i$ or $j > i$. The propagation of bit-flipping errors is limited: a bit flipped in C_i only affects P_i. This error propagation is the same as in the One-Time Pad (OTP) [Menezes et al. 1997]. Note that only E_K is enough for both the encryption and decryption modes.
- CTR: in counter mode each ciphertext block is computed as $C_i = P_i \oplus E_K(X_i)$ where $X_1 =$ IV and $X_i = E_K(f(X_{i-1}))$ for $i > 1$, with f a simple counter function or a Linear Feedback Shift Register (LSFR). CTR is a stream mode, thus, there is no need for padding, and error propagation is limited (like in a OTP). Just like in OFB, only E_K is enough for both the encryption and decryption operations in CTR mode.
- XTS: in XEX Tweakable with ciphertext Stealing mode, each ciphertext block C_i is computed as $C_i = X_i \oplus E_{K_2}(P_i \oplus X_i)$, where $X_i = E_{K_1}(i) \otimes \alpha$. According to [Dworkin 2010a], $n = 128$ bits, E_K is AES, α is a primitive element of GF(2^{128}), and $\otimes$ is multiplication in GF(2^{128}). XTS is a parallelizable mode, operating blockwise like in ECB, but requiring double encryption per block. XTS uses ciphertext stealing [Menezes et al. 1997] when $m \not\equiv 0 \bmod n$.

4. Distinguishing Attacks

The weakest goal of an adversary is to be able to distinguish a ciphertext from a random string. If a cipher does not leak information on the plaintext through to the ciphertext, then adversaries cannot distinguish the given cipher from a random permutation (over the same plaintext space). In this paper, we focus exclusively on this type of distinguishing attack. A modern trend is to complement the confidentiality property with an authentication tag, such as in IACBC (Integrity-Aware CBC) and IAPM (Integrity Aware Parallelizable Mode) [Jutla 2001, Jutla 2000]. There are several authenticated-encryption (AE) modes such as CCM (CBC-MAC with Counter Mode) [Whiting et al.], EAX (uses OMAC) [Bellare et al. 2004], CWC (Carter-Wegman-Counter) [Kohno et al.] and GCM

(Galois-Counter Mode) [McGrew and Viega 2004]. They perform two (or more) passes over the input message, but one pass is for encryption while the other passes are for computing an authentication tag. Our focus is on confidentiality modes only.

Our attacks deal with the dichotomy n versus m, that is, the fact that modes of operation using block ciphers E_K are inevitably bound to operate on n-bit blocks, for fixed n, while random permutations can freely operate on m bits, without need to partition the plaintext in n-bit (or smaller) pieces. Our attacks use very few known- or chosen-plaintext (KP or CP) queries and are independent of the key size, the number of rounds, the block size n and the internal cipher components of E_K.

The classical case $n = m$ has already been treated [Bellare et al. 1997, Bellare and Rogaway 2006]. The motivation to move beyond the setting $n = m$ is that it allows us to view the interaction between different n-bit encrypted blocks. The setting $n \neq m$ is powerful since it allows us to exploit peculiar behaviors of modes of operation (padding, blockwise operation, IV, poor diffusion) that set them apart from random permutations when operating on arbitrary-size plaintext messages.

We focus our analyses on two cases:
(i) $n > m$: in this case, for ECB, XTS and CBC modes, some padding scheme is needed because E_K necessarily operates blockwise and cannot be applied to less than n bits. On the other hand, π^m operates smoothly on m-bit inputs without padding, and generates an m-bit output. For E_K, even ciphertext stealing [Dworkin 2010b] is not an option since there are no previous ciphertext block to steal bits from. Even if bits are stolen from an initial value (IV) or from the key K, the end result is ciphertext expansion: while the input block has m bits, the ciphertext output has necessarily $n > m$ bits for E_K. Moreover, the excess $n - m$ bits cannot be removed otherwise decryption will not work. Therefore, the lenght of the ciphertext alone indicates if E_K or π^m was used, and the advantage in distinguishing between the two will be 1. In the XTS and CBC modes, different messages may use different initial values (IVs), but this is not an issue in our attacks. Exceptionally, in this case, we only need a single known-plaintext query.

We assume that IV's, nonces and tweaks are agreed upon between the legitimate parties like the key K. Nonetheless, the former are public values while K is secret. We assume that the former are not accounted for in the input size n nor in m. In other words, these auxiliary values do not consume bandwidth, i.e. they are not transmitted along with the ciphertext. Otherwise, they would lead to ciphertext expansion and we could use them to discriminate between E_K and π^m (since the latter clearly does not need them).

In OFB and CTR modes, only m keystream bits are enough to encode an m-bit message. These stream modes have the same bitwise diffusion as the One-Time-Pad (OTP): if a single bit of the ciphertext flips, only the corresponding bit of the plaintext flips (after decryption). We query a single, known m-bit message P and obtain the corresponding ciphertext C. Next, we flip a single bit of C to get C' and ask for its decryption. In both OFB and CTR modes, the corresponding plaintext P' from C' will differ in a single bit compared to P and in the same position of the bit changed in C. For π^m, the entire plaintext will be garbled, and the probability that a single bit flip in C leads to a single bit flip in P is $1/2^{m-1}$ since $m - 1$ bits have to be equal to both P and P'. The advantage in this case is $1 - 2^{1-m}$. The larger m is, the larger the advantage. To achieve an even larger advantage, another bit of C could be flipped, leading

to C'', and the attack repeated. We cannot use two messages (P, P') differing in a single bit because each of P and P' would necessarily require different IVs. In CTR mode the counter is the IV.

In b-bit CFB mode, typically $b = 1$ or $b = n$ but let us assume $b = m$ is allowed. Then, the attack is similar to the one in OFB and CTR modes. Let P be a known message with $|P| = m$ bits. The m-bit ciphertext is $C = P \oplus E_K(\text{IV})$. Notice that since the message is smaller than a single block there is no chance of ciphertext chaining, since there is no initial ciphertext, just the IV. Thus, the effect is just like in OFB and CTR modes because $E_K(\text{IV})$ is text-independent.

If $b = 1$, then we encrypt P as before and get C. Next, we flip the last bit of C to get C', and ask for its decryption. The flipped bit of C' will be fedback into the state at the latest and the corresponding plaintext P' will differ from P only in the last bit under E_K. The probability for this single bit difference in π^m is $1/2^{m-1}$ i.e. $m - 1$ bits will have to be equal under π^m. Even if b were secret, there are only a finite number of possibilities for b: 1 up to $m - 1$. With at most $2(m-1)$ queries, the previous attack can be repeated for every possible value of b. The advantage is $1 - 2^{1-m}$.

(ii) $n < m$: there are two subcases to consider

- $m \equiv 0 \bmod n$: the ECB mode is straightforward to analyse. Just query repeated blocks (P, P, P) and observe if the ciphertext is a repeated sequence (C, C, C). If so, then the adversary identified a block cipher E_K, otherwise, a random permutation π^m. The advantage is $1-2^{-n}$. Observe that while E_K operates on n-bit blocks, π rather operates on the entire sequence (P, P, P) at once, and there is only a tiny chance 2^{-n} that the result would be (C, C, C).
  In CBC mode, the adversary asks two queries $(P_1,\ P_2,\ P_3)$ and $(P'_1,\ P_2,\ P_3)$ such that $P_1 \oplus \text{IV} = P'_1 \oplus \text{IV}'$, where IV and IV' are the corresponding initial values [Bellare et al. 1997]. Note that we choose P_1 and P'_1, not the IVs. Thus, $C_1 = E_K(P_1 \oplus \text{IV}) = E_K(P'_1 \oplus \text{IV}') = C'_1$. Since the remaining blocks are the same for the rest of the message, and the first ciphertext block fedback in CBC mode is the same in both messages, the remaining ciphertext blocks are also identical for E_K. For a random permutation on m bits, this collision will never happen since π^m is a permutation. The advantage is 1.
  If (ever) the IV happens to be the same, then we query two messages (P_1, P_2, P_3) and (P_1, P_2, P'_3) such that $P_3 \neq P'_3$. Notice that the (ciphertext) chaining in CBC is in the left-to-right direction. Left-to-right chaining means that P_i is processed before P_j for $i < j$. In summary, P_i blocks are encrypted for increasing values of i starting with $i = 1$. Therefore, P_j depends on P_i for all $i < j$, but not the other way around. Thus, only C_3 will differ: C_1 and C_2 will be the same for both messages since the IV is the same. For π^m, in this case, the probability is 2^{-2n} for two consecutive n-bit blocks to be equal, and the advantage is $1 - 2^{-2n}$. For E_K and the given messages, the two n-bit ciphertext blocks C_1 and C_2 will always be the same. The advantage grows for longer messages.
  In OFB, XTS and CTR modes, we make a message query P and obtain C. Further, we flip a single bit of C to get C', and ask for its decryption. For E_K, just a single bit of the resulting plaintext P' will differ from P like in a One-Time Pad (OTP). For π^m, the probability of observing a 1-bit difference in two m-bit plaintexts is $1/2^{m-1}$, and

the advantage is $1 - 2^{-8}$. Note that in this case the adversary is making an adaptively chosen-ciphertext query, and the decrypted ciphertext results in a meaningfull plaintext (except, eventually, for the garbled bit position). Again, notice that in OFB, XTS and CTR modes there is no plaintext-dependent chaining. Likewise, for b-bit CFB mode, the attack proceeds like in OFB mode since the diffusion is in the left-to-right direction only.

- $m \not\equiv 0 \bmod n$: this case is similar to the case $n > m$, and the focus is on the last message block that contains only $m \bmod n$ bits. The treatment of these trailing bits by each mode of operation allows the adversary to detect whether E_K or π^m was used. For ECB, XTS and CBC modes, ciphertext stealing could be used, and our previous argument in the case $n > m$ do not apply. For ECB and XTS modes, the adversary queries two messages (P_1, P_2, P_3) and (P_1, P_2, P_3') where $|P_3| = |P_3'| = m \bmod n$, but $P_3 \neq P_3'$. For CBC mode, the messages are (P_1, P_2, P_3) and (P_1', P_2, P_3') where $|P_3| = |P_3'| = m \bmod n$, but $P_3 \neq P_3'$. P_1 and P_1' are such that $P_1 \oplus \mathrm{IV} = P_1' \oplus \mathrm{IV}'$, so $C_1 = C_1'$.
  In ECB, XTS and CBC modes, after padding, only C_3 and C_3' will differ while $C_i = C_i'$ for $i < 3$ whatever E_K is used. If the same IVs are ever used, we can just choose different P_3 and P_3'. Thus, the adversary can distinguish between E_K and π^m with advantage $1 - 2^{m \bmod n - m}$ for m-bit messages, since only the last $m \bmod n$ bits differ in both messages.
  For OFB, CTR and CFB modes there is no padding, but the same strategy as in the OTP also apply: we exploit the bitwise diffusion.

In our attacks, we exploited the following facts that are inherent to any block cipher E_K using a confidentiality mode of operation:

- padding and ciphertext stealing: in ECB, XTS and CBC modes, the size of each text block has to be at least n bits, because E_K cannot operate on smaller blocks. To fill in the missing bits, padding is needed. It does not matter which padding scheme is used since there will be ciphertext expansion anyway, and this fact alone is enough to detect that E_K was used instead of π^m. Notice that random permutations π^m never need padding.
- left-to-right (L2R) diffusion and one pass over the message: CBC and CFB modes applied to a message $(P_1, P_2, P_3, \ldots)$ chains values in left-to-right order (and never the other way around), that is, C_i depends on C_j and indirectly on P_j for $j \leq i$, but C_i is independent of C_l and P_l for $l > i$. In a sense, the left-to-right chaining order makes both the CBC and the CFB modes a kind of T-function [Klimov and Shamir 2002]. This unidirectional diffusion is due to the design of these modes: only a single pass is allowed over the message due to efficiency and buffering considerations. We exploited precisely these weaknesses to construct our message queries and attacks. Notice that the attacks work independently of the underlying block cipher E_K or the key size. In comparison, for π^m there is full diffusion across an entire m-bit string. Moreover, the avalanche effect holds for π^m: changing a single bit in any of the m input bits implies all output bits change with 50% chance. For E_K over m-bit messages, the avalanche effect does not hold.
- plaintext-independent chaining: in ECB, XTS, OFB and CTR modes, the dependence between consecutive n-bit blocks (if ever) depends on the key, the tweak or the IV but not on the plaintext nor the ciphertext. This feature is motivated by parallel

processing capabilities of these modes to speed-up encryption. In π^m, we expect full text-dependent diffusion across the entire m-bit string.

- bitwise diffusion in OTP: in streaming modes such as OFB and CTR, the key bit-stream generated simulates a One-Time-Pad in the sense that the ciphertext is simply the message xored to a plaintext-independent key stream. This fact means that diffusion is worse than the left-to-right diffusion pointed out for the CBC and CFB modes: if only a limited set of bits change in the message, the very same isolated set of bits will change in the ciphertext (and vice-versa). This is extremely unlikely to be observed in a random permutation π^m operating on the whole m bits at once, and this phenomenon can be detected for E_K with only two queries: one encryption and one decryption.

The probability of the adversary simply guessing whether the oracle he interacts with is E_K or π^m, is 1/2. In all cases, our attacks have advantage much larger than 1/2, for appropriate and realistic values of m.

In summary, all the modes analysed previously leak information, that is, leave footprints or signatures of their presence in the ciphertext, independently of which block cipher E and key K are used. For instance, a random permutation π^m provides full diffusion across an m-bit string as a monolithic transformation. On the other hand, all modes of operation mentioned necessarily work blockwise, n bits at a time, and in the left-to-right direction, i.e diffusion is unidirectional. Thus, the avalanche effect is compromised.

To fix these problems, we claim that:

- achieving complete diffusion is necessary; it is suggested that modes of operation perform two passes over the m-bit message in both left-to-right (L2R) and right-to-left (R2L) directions. L2R is the natural order in which P_i blocks are presented in the input: P_i before P_j for $i < j$. Therefore, L2R difusion means that P_j depends on P_i for $j > i$, but not the other way around. R2L means the opposite, i.e. block P_i is processed before P_j for $i > j$. Separately, L2R and R2L provide weak diffusion, but combining L2R and R2L results in much stronger diffusion.
  A drawback with two passes over the message is buffering: the intermediate data processed in the first pass should be securely stored[2] for the second pass (in the reverse direction), before ciphertext is output. Well-known modes of operation such as PEP [Chakraborty and Sarkar 2006], CMC [Halevi and Rogaway 2003] and EME [Halevi and Rogaway 2004] already required buffering due to multiple passes over the data. The buffering issue is less critical in settings such as in disk-sector encryption [SISWG] since only 512 bytes need to be stored, which is a small amount and is known beforehand. In general, though, the total size of the input, m, is not known in advance. If the intermediate data, for example, $X_i = E_K(P_i \oplus X_{i-1})$ in a 2-pass mode is leaked, then the n-bit secret intermediate state X_i is exposed and security may be compromised [Biham 1998], for example, by a meet-in-the-middle attack. The fact that hard-disk encryption modes actually use multiple passes means that our attacks are relevant.
- modes should use chaining that is either plaintext or ciphertext dependent, such as in CBC and CFB modes. Multiple passes, in opposite directions, over the data for modes such as ECB, OFB and CTR are void, since these modes have no text-dependent chaining. For instance, 2-pass CTR mode (with or without the same IV or key) still does

[2]We assume some kind of secure storage is available. It is intended to protect the partially (1-pass) intermediate encrypted data from leaking.

not counter the attacks described previously since xoring two key streams (under different counters and keys) are equivalent to applying two OTP keystreams in succession. In other words, diffusion remains bitwise in both 1-pass and 2-pass CTR because the key streams are independent of plaintext and ciphertext. In fact, the same reasoning holds for any number of passes of CTR mode. The same rationale applies to ECB and OFB modes. A drawback of our recommendation is that (chained) modes become non-parallelizable due to text-dependent chaining. Another consequence of the two-pass procedure is that text-dependent chaining causes infinite error propagation across the entire m-bit ciphertext. This effect, though, simply means complete diffusion was achieved.

- finally, to deal with both the cases $n < m$ and $n > m$ a stream mode should be used. For $n < m$, there are padding schemes, but for $n > m$ there is no way out for modes that operate blockwise, such as in ECB and CBC.

These conditions are aimed to make the modes behave closer to a random permutation over m-bit strings.

5. Two-pass modes

Let an m-bit input message be denoted $P = (P_1, P_2, P_3, \ldots, P_t)$, where $|P_i| = n$ bits for $1 \leq i \leq t-1$, $|P_t| = n - m \bmod n$ bits and $t = \lceil m/n \rceil$. We assume randomly chosen, uniformly distributed, publicly-known n-bit initial values, IV_j for $j > 0$, if needed. Conventionally, 2-pass modes have been associated with authentication modes such as AES-CCM [Dworkin 2004], but, in the later, the second pass is aimed at computing an authentication tag on top of the confidentiality service of a mode of operation. Therefore, even in AES-CCM, diffusion is still unidirectional: both passes over the data are in left-to-right direction and thus, do not provide appropriate diffusion across an m-bit block.

There are 2-pass modes of operation that perform R2L diffusion, such as the EMD (Encrypt-Mask-Decrypt) and CMC (CBC-Mask-CBC) modes [Rogaway , Halevi and Rogaway 2003]. But, they use CBC decryption in right-to-left order, and they are aimed at encryption of fixed-size disk sectors, not arbitrary m-bit messages. It means that these modes are allowed only if $m \equiv 0 \bmod n$, since no padding scheme was defined. Moreover, the CMC and EMD modes contain an additional masking layer in between the two CBC layers. The mask, denoted M in [Halevi and Rogaway 2003], mixes intermediate blocks X_i using multiplication over GF(2^n). This mask provides diffusion across n-bit blocks and helps counter exhaustive search attacks on specific n-bit blocks. Without M, diffusion would be much weaker: suppose we have two messages $(P_1, P_2, P_3, P_4, P_5)$ and $(P_1, P_2, P_3, P_4, P'_5)$ that differ only in the last block: $P_5 \neq P'_5$. Then, during the first pass in CBC, only X_5 is different from X'_5. Since there is no mask, X_5 and X'_5 only affect C_5, C'_5 (directly) and C_4, C'_4 (due to the backwards text chaining in the CBC decryption of the second pass). The rest of the ciphertext is not affected and diffusion is limited. See Fig. 1.

An alternative scheme would be to perform two-pass CBC both in left-to-right direction, but chaining X_t as IV to X_1 in the second pass since X_t would depend on all P_i. Thus, $C_1 = E_K(X_1 \oplus X_t)$ and, for $1 < i \leq t$, we have $C_i = E_K(X_i \oplus C_{i-1})$. We call it wrapped-CBC mode. See Fig. 2(a). For simplicity, let us consider the $m \equiv 0 \bmod n$ case. There is full diffusion across an m-bit message due to the chaining of X_t; the intermediate blocks X_i mask the P_i and also protect (P_1, C_1) from exhaustive key search attacks. Nonetheless, the

decryption mode is quite weak because all chainings are only fed forward: changing C_1, for instance, will affect P_1, P_2 and P_3 only. See Fig. 2(b).

To avoid this kind of discrepancy between the encryption and decryption modes, an alternative is 2-pass IGE mode. The 1-pass IGE mode was described by C. Campbell in [Campbell 1978]. IGE means Infinite Garble Extension because a single bit flipping error causes infinite error propagation, that is, all subsequent blocks, including the one with the error, will be garbled due to chaining in encryption/decryption. Still, 2-pass IGE mode cannot deal with the case $n > m$ without causing ciphertext expansion, since encryption is blockwise like in ECB and CBC modes. Therefore, only stream modes can deal with all m values.

For completeness sake, we describe the 2-pass IGE encryption which we call wrapped-IGE mode, and is depicted in Fig. 3(a). We assume $n < m$. There are two sub-cases to consider

- $m \equiv 0 \bmod n$: in this case, all blocks are n bits wide. In the first pass over the m-bit message, the conventional IGE mode is applied and intermediate blocks X_i are generated. Diffusion only occurs in the left-to-right direction: $X_1 = E_K(P_1 \oplus \text{IV}_1)$, and for $1 < i \leq t$ we have $X_i = E_K(P_i \oplus X_{i-1}) \oplus P_{i-1}$. Note the chaining from the previous blocks P_{i-1}, and the unknown X_{i-1} masks the P_i block, hiding it. For the second pass, the last ciphertext block is initially generated as $C_t = E_K(X_t) \oplus \text{IV}_2$, and for $1 \leq i < t$ we have $C_i = E_K(X_i \oplus C_{i+1}) \oplus X_{i+1}$ in the reverse direction. Note the feedforward of C_{i+1} leading to avalanche in the in the right-to-left direction. The intermediate X_i blocks mask the outputs of E_K.
  Unlike wrapped-CBC mode, the decryption of wrapped-IGE mode has the same (complete) diffusion across the m-bit block as encryption. An inconvenience is that the ciphertext blocks must be processed in reverse order: from C_t down to C_1. See Fig. 3(b).
  Unlike CMC and EMD modes, wrapped-IGE does not use nor need multiplication over GF(2^n), since the chaining of X_i and C_i values are enough to provide the necessary diffusion.
- $m \not\equiv 0 \bmod n$: in this case, there is a final block with only $m \bmod n$ bits that will be treated with the ciphertext-stealing technique [Menezes et al. 1997] to avoid ciphertext expansion.
  In the first pass, we proceed as before in IGE mode, but taking care of the last partial block. Again, intermediate data X_i is generated: $X_1 = E_K(P_1 \oplus \text{IV}_1)$ and, for $1 < i < t$ we have $X_i = E_K(P_i \oplus X_{i-1}) \oplus P_{i-1}$. For the last partial block we use ciphertext stealing. Recall that $|P_t| = m \bmod n$ bits, so $X_t = E_K((P_t \| \text{lsb}_{n-m \bmod n}(X_{t-1}) \oplus X_{t-1}) \oplus P_{t-1}$, where $\text{lsb}_v(y)$ denotes the v least significant bits of y. To adjust the size of the output to the same size m of the input message, we rearrange the bits as follows: $X_j = \text{lsb}_{n-m \bmod n}(X_{j-1}) \| (X_j \gg n - m \bmod n)$ for $1 < j \leq t-1$ and $X_t = \text{lsb}_{m \bmod n}(X_t)$, where $x \gg y$ denotes x shifted right by y bits (the y least significant bits of x are dropped). There are two reasons for the rearrangement of bits: (i) the second pass explained in the next paragraph; (ii) the $\text{lsb}_{n-m \bmod n}(X_{t-1})$ was used in (and can be recovered from) X_t, there is no need to keep it in both X_t and X_{t-1}. Notice that $\text{lsb}_{n-m \bmod n}(X_1)$ became redundant.
  For the second pass, we move in the right-to-left direction, guaranteeing full diffusion across the entire m-bit string. Initially, $C_t = E_K(X_t \oplus \text{IV}_2)$. For $1 < i < t$, we have $C_i = E_K(X_i \oplus C_{i+1}) \oplus X_{i+1}$. Finally, for the last block, we have $C_1 =$

$E_K((X_1\|\text{msb}_{m \bmod n}(C_2)) \oplus C_1) \oplus X_1$. Lasty, we adjust the size of the ciphertext: $C_2 = \text{lsb}_{n-m \bmod n}(C_2)$. This way, there is no ciphertext expansion.

Now, for a secure streaming mode, we suggest 2-pass b-bit CFB or wrapped b-bit CFB. See Fig. 4. We assume n-bit initial values IV_i, for $i > 0$, as needed. We assume that k-bit keys K_1 and K_2 are dependent, for instance, jointly generated like $K_1 = E_K(S)$ and $K_2 = E_K(S \oplus K_1)$ from a random n-bit seed S and a random k-bit key K, with $|S| = |K| = |K_1| = |K_2|$. For instance, E_K is AES with $n = k = 128$. This requirement on K_1 and K_2 aims to counter meet-in-the-middle attacks.

The encryption proceeds as follows: (i) $n > m$: we cannot use $b = m$, even though that would be more efficient than a smaller b. Recall that the keystream generated by CFB mode is exclusive-ored to P, but if the keystream itself depends on P, then, there is a self-referential issue: $C = P \oplus \text{msb}_m(E_{K_1}(\text{msb}_{n-m}(\text{IV}_1)\|P))$ cannot be decrypted.

Even if we encrypt $P\|0^{n-m}$ i.e. P padded with $n - m$ zero bits, in the first pass, as $X = (P\|0^{n-m}) \oplus E_K(\text{IV}_1)$. Then, in a second pass, $C = \text{msb}_m(X \oplus (E_{K_2}(X)))$. But, again, due to a self-referential result, we cannot decrypt C. Moreover, E_{K_2} depends on n bits of X, and we only have m bits. Therefore, we assume $b = 1$. Let uppercase symbols, such as S denote an n-bit block while lowercase symbols such as s_i denote a single bit. In the first pass, there is an initialization step: $S = E_{K_1}(\text{IV}_1) = (s_1, \ldots, s_n)$. Next, the bits of $P = (p_1, \ldots, p_m)$ are encrypted one by one, and the result is fedback into E_{K_1}: $x_i = p_i \oplus \text{lsb}_1(S)$, $S = E_{K_1}((S \ll 1)\|x_i)$, for $1 \leq i \leq t$, where $x \ll y$ denotes x left-shifted by y bits. There is no need for padding, since encryption operates bitwise. Using $b = 1$ is inefficient, but since $m < n$, the penalty is minimal. The result is an m-bit string $X = (x_1, \ldots, x_m)$.

In the second pass, we wrap around. Initially, $Y = E_{K_2}(\text{lsb}_{n-m}(\text{IV}_2)\|X_t) = (y_1, \ldots, y_m)$. Then, $c_{m-i} = y_i \oplus \text{lsb}_1(Y)$, $Y = E_{K_2}((Y \ll 1)\|c_{m-i})$, for $1 \leq i \leq t$. Note the indexing of the ciphertext bits c_{m-i}, for $1 \leq i \leq t$, indicating 'right-to-left' direction. This case shows how a streaming mode such as CFB can deal efficiently and smoothly with variable-length inputs, while IGE and other blockwise modes could not, particularly the $n > m$ case, even with padding.

(ii) $n < m$: there are two subcases to consider:

- $m \equiv 0 \bmod n$: in this case, we use $b = n$ which means full feedback to improve performance. In the first pass over the m-bit message, $Y = E_{K_1}(\text{IV}_1)$ is initially generated. Diffusion occurs in the left-to-right direction following a blockwise encryption: $X_i = P_i \oplus Y$ and $Y = E_{K_1}(X_i)$, for $1 \leq i \leq t$. For the second pass, we wrap-around: initially, $Y = E_{K_1}(X_t)$. Then, repeatedly $C_i = X_i \oplus Y$ and we update $Y = E_{K_2}(C_i)$, for $1 \leq i \leq t$. We could alternatively have reversed direction: $C_{t-i} = X_{t-i} \oplus Y$ and $Y = E_{K_2}(C_{t-i})$ for $t > i > 0$. Both options are equivalent.
- $m \not\equiv 0 \bmod n$: we have two choices: (i) we can use $b = 1$, which is inefficient for large values of m and fixed values of n; (ii) we can use $b = n$ for $\lfloor m/n \rfloor$ blocks (most of them) and then switch to $b = 1$ for the last partial block of $m \bmod n$ bits. Option (ii) is similar to the item $n > m$ where encryption is bitwise using a single bit from E_{K_j}, $j \in \{1, 2\}$, at a time. Option (i) is a mixture of the items $m \equiv 0 \bmod n$ (for $m > n$) and $n > m$ (for the last $m \bmod n$ bits). Again, there are two passes over the m-bit message in opposite directions: left-to-right and right-to-left.

The wrapped-CFB mode counters all the attacks described against 1-pass modes, for

both $n < m$ and $n > m$. Flipping a single or even multiple bits of the ciphertext C_i (resp. plaintext P_i) will affect all plaintext bits of P_i (resp. ciphertext C_i) due to text chaining and bi-directional diffusion in the intermediate blocks X_i. The double pass in opposite directions guarantees full diffusion for both encryption and decryption, making them equally strong. Concerning meet-in-the-middle (MITM) attacks, the first n-bit block P_0 is the most interesting target: $C_0 = P_0 \oplus E_{K_1}(IV_1) \oplus X_0$ and $X_0 = C_0 \oplus E_{K_2}(C_1)$. The other pairs (P_j, C_j), $j > 0$, contain an unknown quantity X_{j-1}. In a known-plaintext setting P_0 is known. IV_1, C_0 and C_1 are also known. If K_1 and K_2 were independent k-bit keys, then a MITM attack could be applied to the (P_0, C_0) pair with time complexity around 2^k calls to E_K, instead of 2^{2k} [Menezes et al. 1997]. But, by construction, K_1 and K_2 are (nonlinearly) dependent.

What about birthday-paradox-type attacks for m-bit messages? For random, unpredictable IVs and keys, there is no collision possible in wrapped-CFB mode because this mode effectively performs an m-bit permutation. It means that two distinct m-bit plaintexts (resp. ciphertexts) always lead to different m-bit ciphertexts (resp. plaintexts). Consider now the case of variable IVs. Since the text-independent parameters consist of $(\mathrm{IV}_1, \mathrm{IV}_2, K_1, K_2)$, then if the keys are fixed and only the IVs change then, a birthday-paradox effect on $(\mathrm{IV}_1, \mathrm{IV}_2)$ would lead to a collision in the m-bit ciphertexts after $\sqrt{2^m}$ encryptions under 2^{2n} different IVs (assuming $|\mathrm{IV}_1| = |\mathrm{IV}_2| = n$). This means $m \approx 2n$, but if we assume $n = k$, then an effort of $\sqrt{2^m}$ encryptions is the same as an exhaustive key search. If $m > 2n$, then collisions are void.

There are several modes of operation that aim to achieve better diffusion than the modes in [Dworkin 2001, IEEE 2008]. Multiples encryption passes over the data cannot be avoided if full diffusion is the objective. A strategy is to achieve full diffusion by employing so called universal hash functions instead of encryption. For instance, the Hash-Encrypt-Hash (HEH) mode [Sarkar 2007] was based on the Naor-Reingold [Naor and Reingold , Naor and Reingold 1999] paradigm. HEH targeted disk-sector encryption where the input message is a single disk sector. The buffering issue is minimized, since for disk-sector encryption the storage needed is only $m = 4096$ bits or 512 bytes. The HEH mode uses the ECB mode between two layers of a universal hash function $H : \mathbb{Z}_2^t \rightarrow \mathbb{Z}_2^t$ (thus, the name HEH). Moreover, m must be a multiple of n, the input size of E_K. Diffusion across an m-bit string is provided by the hash function H, which is invertible and cheaper to compute than several E_K instances. But, if m is not a multiple of n, it is straightforward to distinguish E_K in HEH mode from π^m since HEH cannot be applied due to padding issues.

6. Conclusions

In this paper, we argued about **limitations/drawbacks in six standard confidentiality modes of operation: ECB, CBC, OFB, CFB, CTR and XTS that perform a single pass over the input message**. A pervasive problem is unidirectional diffusion (left-to-right direction) or bitwise diffusion (in stream modes: OFB, CFB and CTR). Similar conclusions hold for the inverse of these modes, whatever the underlying n-bit block cipher E_K and whatever the key K. To compound the problem, it is rare to find text-dependent chaining (only present in CBC and CFB modes). Consequently, these modes behave significantly worse than a random permutation over message spaces larger or smaller than a single n-bit block. Therefore, the 1-pass modes cannot properly model a random permutation over message spaces composed of m-bit strings.

We claim that using the message space of m bits is more relevant as a testing ground,

since in practice messages do not respect n-bit boundaries. This setting has already been discussed and is widely accepted for disk-sector encryption, but in the latter m is limited to 512 bytes, which is a multiple of n for many block ciphers such as the AES. In this paper, we allow m to be unrelated to n, not necessarily a multiple of n, although at most a polynomial in n.

The distinguish-from-random attacks described in this paper can be countered by processing the entire message in two passes in opposite directions: left-to-right and right-to-left, to provide full diffusion across the m-bit input. Moreover, this countermeasure requires text-dependent chaining. Also, to account for the case $n > m$, streaming modes are necessary, since any padding would cause ciphertext expansion. This combination of double-pass and text-dependent chaining guarantees complete diffusion just as random permutations would do and as would be expected of a block cipher aimed at mimicking the behaviour of π^m over large m-bit strings.

In our attacks, the adversary may need black-box access to both encryption and decryption oracles. The queries are small in size (a few n-bit blocks each) and we used at most two message queries. In all cases, the amount and size of queries are polynomial-sized in n. Our attacks have high advantage and are independent of n, or of the cipher structure (Feistel, SPN, IDEA-like, LFSR-like) or the key size or the number of rounds. Therefore, our attacks apply independently of the underlying block cipher. Table 1 summarizes the results in this paper. From this table, we conclude that under the given assumptions on (n, m), no block cipher in ECB mode can be either IND-KPA or IND-CPA. Analogously, no block cipher in CBC mode can be either IND-KPA or IND-CPA; no block cipher in CFB mode can be either IND-KPA or IND-CCA2; no block cipher in CTR mode can be either IND-KPA or IND-CCA2; no block cipher in OFB mode can be either IND-KPA or IND-CCA2; no block cipher in XTS mode can be either IND-KPA or IND-CPA.

References

Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997). A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS'97*, pages 394–403.

Bellare, M. and Rogaway, P. (2006). The security of triple encryption and a framework for code-based game-playing proofs. In Vaudenay, S., editor, *Adv. in Cryptology, Eurocrypt*, volume 4004 of *LNCS*, pages 409–426. Springer.

Bellare, M., Rogaway, P., and Wagner, D. (2004). The eax mode of operation. In *Fast Software Encryption (FSE)*, volume 3017 of *LNCS*, pages 389–407. Springer.

Biham, E. (1998). Cryptanalysis of multiple modes of operation. *Journal of Cryptology*, 11(1):45–58.

Campbell, C. (1978). Design and specification of cryptographic capabilities. In Brandstad, D., editor, *Computer Security and the Data Encryption Standard*, Special Publications 500-27, pages 54–66. National Bureau of Standards, US Dept of Commerce.

Chakraborty, D. and Sarkar, P. (2006). A new mode of encryption providing a tweakable strong pseudorandom permutation. In *Fast Software Encryption (FSE)*, volume 4047 of *LNCS*, pages 293–309. Springer.

Courtois, N. (2006). How fast can be algebraic attacks on block ciphers? IACR ePrint archive 2006/168.

Table 1. 1-pass modes, attack complexities, advantage and weaknesses.

1-pass mode	attack complexity data/memory†	attack complexity time	advantage	issue	comments
ECB	1 KM	1	1	padding	$n > m$
ECB	1 CM	1	$1 - 2^{-n}$	blockwise diffusion	$n < m, m \equiv 0 \bmod n$
ECB	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m, m \not\equiv 0 \bmod n$
CBC	1 KM	1	1	padding	$n > m$
CBC	1 CM	1	1	collision	$n < m, m \equiv 0 \bmod n$
CBC	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m, m \not\equiv 0 \bmod n$
CFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bit diffusion	$n > m$
CFB	1 KM	1	$1 - 2^{1-m}$	L2R diffusion	$n < m, m \equiv 0 \bmod n$
CFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \not\equiv 0 \bmod n$
CTR	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n > m$
CTR	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \equiv 0 \bmod n$
CTR	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \not\equiv 0 \bmod n$
OFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n > m$
OFB	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \equiv 0 \bmod n$
OFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \not\equiv 0 \bmod n$
XTS	1 KM	1	1	padding	$n > m$
XTS	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m, m \equiv 0 \bmod n$
XTS	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m, m \not\equiv 0 \bmod n$

KM: Known Message; CM: Chosen Message; CC: Chosen Ciphertext

†: memory complexity is the space needed to store the given data.

Crowley, P. (2000). Mercy: a fast large block cipher for disk sector encryption. In Schneier, B., editor, *Fast Software Encryption (FSE)*, volume 1978 of *LNCS*, pages 49–63. Springer.

Daemen, J. and Rijmen, V. (2000). The block cipher bksq. In Quisquater, J.-J. and Schneier, B., editors, *Third International Conference on Smart Card Research and Applications (CARDIS)*, volume 1820 of *LNCS*, pages 236–245. Springer.

Dworkin, M. (2001). Recommendation for block cipher modes of operation methods and techniques. National Institute of Standards and Technology NIST Special Publication 800-38A (2001).

Dworkin, M. (2004). Recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality. National Institute of Standards and Technology (NIST). NIST Special Publication 800-38C (2004).

Dworkin, M. (2010a). Recommendation for block cipher modes of operation: The xts-aes mode for confidentiality on storage devices. National Institute of Standards and Technology (NIST). NIST Special Publication 800-38E (2010).

Dworkin, M. (2010b). Recommendation for block cipher modes of operation: Three variants of ciphertext stealing for cbc mode. National Institute of Standards and Technology (NIST). Addendum to NIST SpecialPublication 800-38A (2010).

FIPS197 (2001). Advanced encryption standard (aes). FIPS PUB 197 Federal Information

Processing Standard Publication 197, U.S. Department of Commerce (2001).

Halevi, S. and Rogaway, P. (2003). A tweakable enciphering mode. In Boneh, D., editor, *Adv. in Cryptology, Crypto*, volume 2729 of *LNCS*, pages 482–499. Springer.

Halevi, S. and Rogaway, P. (2004). A parallelizable enciphering mode. In *CT-RSA*, volume 2964 of *LNCS*, pages 292–304. Springer.

IEEE (2008). The xts-aes tweakable block cipher - an extract from ieee std 1619-2007. The Institute of Electrical and Electronics Engineers, Inc.

Jutla, C. (2000). Parallelizable encryption mode with almost free message integrity. http://citeseer.ist.psu.edu/jula00parallelizable.html.

Jutla, C. (2001). Encryption modes with almost free message integrity. In Pfitzmann, B., editor, *Adv. in Cryptology, Eurocrypt*, volume 2045 of *LNCS*, pages 529–544. Springer.

Klimov, A. and Shamir, A. (2002). A new class of invertible mappings. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *LNCS*, pages 470–483. Springer.

Kohno, T., Viega, J., and Whiting, D. Cwc: a high-performance conventional authenticated encryption mode. Cryptology ePrint Archive, report 2003/106 (2003).

Luby, M. and Rackoff, C. (1988). How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386.

McGrew, D. and Viega, J. (2004). The security and performance of the galois/counter mode (gcm) of operation. In Canteaut, A. and Viswanathan, K., editors, *Indocrypt*, volume 3348 of *LNCS*, pages 343–355. Springer.

Menezes, A., van Oorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.

Naor, M. and Reingold, O. A pseudorandom encryption mode. Manuscript available at http://www.wisdom.wiezmann.ac.il/~naor.

Naor, M. and Reingold, O. (1999). On the construction of pseudorandom permutations: Luby-rackoff revisited. *Journal of Cryptology*, 12(1):29–66.

Rogaway, P. The emd mode of operation (a tweaked, wide-blocksize strong prp). Cryptology ePrint Archive 2002/148.

Rogaway, P. (2004). Efficient instantiations of tweakable block ciphers and refinements to modes ocb and pmac. In Lee, P., editor, *Adv. in Cryptology, Asiacrypt*, volume 3329 of *LNCS*, pages 16–31. Springer.

Sarkar, P. (2007). Improving upon the tet mode of operation. In Nam, K.-H. and Rhee, G., editors, *Information Security and Cryptology (ICISC)*, volume 4817 of *LNCS*, pages 180–192. Springer.

SISWG. Ieee security in storage working group (siswg). http://www.siswg.com.

Whiting, D., Housley, R., and Ferguson, N. Submission to nist: Counter with cbc-mac (ccm) aes mode of operation. Computer Security Division, Computer Security Resource Center (NIST).

A. Modes of Operation Schematics

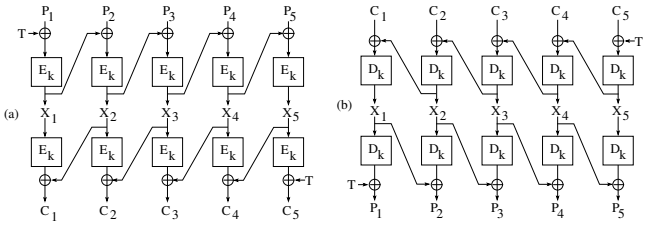


Figure 1. CMC mode without **masking layer (T is tweak): (a) encryption, (b) decryption.**

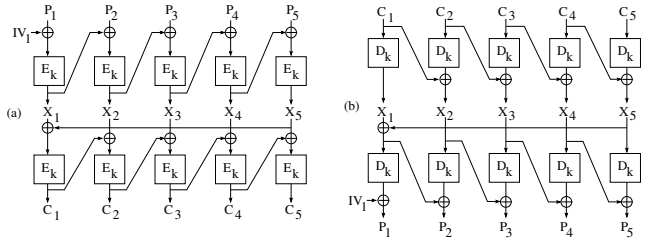


Figure 2. Wrapped CBC mode (2-pass CBC with feedback of last block): (a) encryption, (b) decryption.

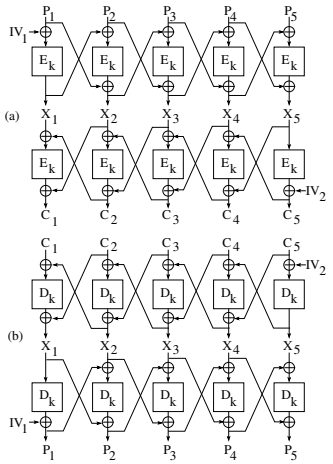


Figure 3. Wrapped IGE mode (two IGE passes): (a) encryption, (b) decryption.

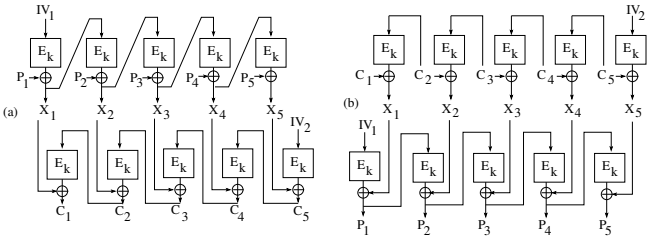


Figure 4. Wrapped CFB mode: (a) encryption, (b) decryption.

Efficient variants of the GGH-YK-M cryptosystem

João M. M. Barguil[1] **, Renan Yuri Lino**[1] **, Paulo S. L. M. Barreto**[1*]

[1] Escola Politécnica, University of São Paulo.

{jbarguil,rlino,pbarreto}@larc.usp.br

Abstract. *The Goldreich-Goldwasser-Halevi (GGH) public-key encryption scheme was deemed broken until recently proposed variants were shown to thwart all known attacks. However, the associated key sizes and generation times are notoriously inefficient. In this paper, we improve on the most promising such variant, proposed by Barros and Schechter and called GGH-YK-M, by reducing public key sizes rom $O(n^2 \lg n)$ down to $O(n \lg n)$ bits, and making key generation over 3 orders of magnitude faster than the results in the literature.*

Keywords: *lattice-based encryption.*

1. Introduction

There is a rising, medium to long term concern with the potential technological viability of quantum computers, because traditional cryptosystems based on the assumed hardness of integer factorization or discrete logarithm computation can be attacked with the help of this new kind of equipment [22]. New schemes based on different computational problems are thus necessary to address this concern, leading to the development of purely classical, but quantum-resistant constructions dubbed *post-quantum* cryptosystems [2].

The most popular family of post-quantum cryptosystems is that of schemes whose security relates to certain hard problems on lattices. Two examples of such problems are the *Shortest Vector Problem* (SVP) and *Closest Vector Problem* (CVP). The former consists of finding a certain approximation (to a factor $\gamma(n)$ where n is the lattice dimension) to the shortest vector in a given lattice, and the latter is to find a lattice vector that is closest to a given vector not necessarily in the lattice. Both of these problems are deemed hard to solve for the Euclidean norm and suitably chosen $\gamma(n)$, as there is no known method to solve them in polynomial time.

One of the pioneering lattice-based encryption schemes, proposed by Goldreich, Goldwasser and Halevi [10] and appropriately dubbed GGH, can be seen as a generalization of the McEliece scheme [16]. In this scheme, a message is translated to a vector in a given lattice and a small error is added. The message is recovered by solving the CVP in that lattice. Known algorithms for solving the CVP work well with short lattice bases, but not with long bases. GGH is a public-key encryption scheme that uses a good lattice basis as the private key and the corresponding Hermite normal form (HNF) [5, section 2.4.2] as the public key. Nguyen proved that the original GGH had inherent structural flaws [19]

*This research was supported by Intel Research grant "Energy-efficient Security for SoC Devices – Asymmetric Cryptography for Embedded Systems" 2012, and by the São Paulo Research Foundation (FAPESP) grant 2013/25977-7. P. Barreto is also supported by the Brazilian National Council for Scientific and Technological Development (CNPq) research productivity grant 306935/2012-0.

and was able to break typical, realistic GGH instances by using lattice reduction algorithms like LLL [12] and BKZ [21].

For several years the GGH scheme was deemed irretrievably broken, to the extent that other kinds of lattices stemming from the Learning with Errors (LWE) problem [20] have essentially dominated the research in the area. This situation began to change when Yoshino and Kunihiro [25] described a variant of GGH (aptly called GGH-YK) that thwarts all known attacks. However, their scheme was incomplete in the sense that, by blindly following their prescriptions, no proper parameter set can be feasibly constructed.

Recently, Barros and Schechter [6] revisited the GGH-YK construction, and proposed a surprising modification of that scheme (dubbed GGH-YK-M, from the fact that it makes essential use of M-matrices [1]) that effectively yields a suitable parametrization. The result is very promising, as it brings the simplicity of GGH and GGH-YK back to life.

The remaining aspect to address, therefore, is to circumvent the inherent high bandwidth occupation and computational cost incurred by all traditional variants of GGH, which make this family of schemes less competitive in practice with other lattice-based encryption methods like Lindner-Peikert [13]. The obvious way to obtain shorter keys in other lattice-based settings like LWE or NTRU [11], namely, resorting to certain rings of structured (e.g. circulant or negacyclic) matrices, fails for GGH because mapping the private key to a public key, that is, computing the HNF, ends up destroying the underlying structure that would enable the size reduction, and thus does not help in attaining that goal.

The technique proposed by Smart and Vercauteren [23] and perfected by Gentry and Halevi [9], targeted at homomorphic encryption, can be used to address this problem. However, the former depends on the lattice determinant to be prime, while the latter relies heavily on the special form of the ring $\mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2. Besides, it requires the computation of resultants and the explicit extraction of the roots of polynomials modulo the lattice determinant, which is done through a quite complex modification of the extended Euclidean algorithm.

Contributions: In this paper we describe an efficient key generation technique that reduces public key bandwidth occupation by an order of complexity, specifically, from $O(n^2 \lg n)$ down to $O(n \lg n)$, while avoiding the need to resort to a full-fledged HNF algorithm, in the same way as the Smart-Vercauteren and Gentry-Halevi methods[1]. Our work extends their technique to any value of n and also for the circulant ring $\mathbb{Z}[x]/(x^n - 1)$, for which we also provide a structural security analysis. In particular, and surprisingly, prime values of n are observed to lead to faster key generation, despite the unavailability of fast Fourier transform techniques to speed up the computations. Our technique only requires a straightforward application of the usual extended Euclidean algorithm, coupled with the

[1]Note added in revision: we were first made aware of the Smart-Vercauteren and Gentry-Halevi key generation techniques after this paper was written. We missed them apparently because of our different target (conventional rather than homomorphic encryption). However, as we explicitly indicate, our proposal is more general, arguably simpler, and empirically more efficient than those methods.

Chinese remainder theorem and the fast Fourier transform.

Our proposal attains much faster processing in all operations involved in a GGH-style cryptosystem, that is, key generation, encryption, and decryption. By far the most pronounced improvement is in key generation, which becomes more than 3 orders of magnitude faster than published results, while encryption becomes almost 2 orders of magnitude faster (our implementation is twice as fast as the literature for decryption). Although our goal was to optimize the GGH-YK-M scheme, it may turn out that our proposal is useful for other scenarios as well, like the somewhat homomorphic encryption scheme of Loftus *et al.* [14] which is the only such scheme so far that resists key recovery attacks [4].

The remainder of this paper is organized as follows. Section 2 introduces basic concepts and notation. We describe the GGH-YK-M scheme in Section 3. Our proposed improvements are put forward in Section 4. In Section 5 we make some security considerations on the improved scheme. The results of experimental assessment and comparisons with the previous state of the art are detailed in Section 6. We conclude in Section 7.

2. Preliminaries

Vector and matrix indices are numbered starting from 0 throughout this paper. We denote by $M_{(i)}$ the i-th row of a matrix M, and by M_j the j-th element on its first row, i.e. $M_j := M_{(0],j}$. We also denote by $x \xleftarrow{\$} U$ the uniformly random sampling of variable x from set U.

Definition 1. *Let $P \in \mathbb{C}^{n\times n}$. The* spectral radius *of P is the quantity $\rho(P) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } P\}$.*

Definition 2. ([1, Definition 1.2]) *Let $P \in \mathbb{Z}^{n\times n}$ such that $P_{ij} \leqslant 0$ for all $0 \leqslant i, j < n$. A (nonsingular) M-*matrix *is a matrix of form $A = \gamma I + P$ for some $\gamma > \rho(P)$.*

Definition 3. ([5, section 2.4.2]) *A matrix $H \in \mathbb{Z}^{n\times n}$ is said to be in* Hermite normal form *(HNF) if it is upper triangular, all its elements are non-negative and the entries on the diagonal are positive and are the largest entries in their respective columns.*

Definition 4. *A matrix $H \in \mathbb{Z}^{n\times n}$ in HNF is said to be* minimal *if it has the form*

$$H = \left[\begin{array}{c|c} I_{n-1} & v^T \\ \hline 0^{n-1} & d \end{array}\right],$$

where $v \in \mathbb{Z}^{n-1}$ and $d \in \mathbb{Z}$.

One can check by direct inspection that the inverse (over $\mathbb{Q}$) of a matrix H in minimal HNF is

$$H^{-1} = \left[\begin{array}{c|c} I_{n-1} & -(1/d)v^T \\ \hline 0^{n-1} & 1/d \end{array}\right].$$

Thus a matrix H in minimal HNF can be conveniently represented by $(v, d) \in \mathbb{Z}^n$ alone. Also, it is clear that $\det(H) = d$.

3. The GGH-YK-M scheme

We now summarize the intriguing GGH variant proposed by Barros and Schechter [6], which itself improves on the GGH-YK scheme by Yoshino and Kunihiro [25], and was called GGH-YK-M by virtue of resorting to M-matrices [1] to complete the specification of that scheme.

For simplicity and efficiency, in our description of GGH-YK-M we explicitly require that the private lattice basis A be such that its HNF is minimal.

Let n be an integer (usually, but not necessarily, a power of 2), let γ be a multiple of n by some small factor (i.e. $\gamma = \alpha n$ for some small integer α), let σ be an even integer, and let h and k be integers such that $h + k < \gamma < 2h$. The GGH-YK-M encryption scheme [6] was designed to thwart all known attacks applicable against the GGH scheme [10], and consists of the following three algorithms:

- Keygen: Sample $P \xleftarrow{\$} \{-1, 0\}^{n \times n}$, compute $A \leftarrow \gamma I + P$ and its HNF $H := \mathrm{HNF}(A)$ until $\rho(P) < \gamma$, $1/\gamma < \left|(A^{-1})_{ii}\right| \leqslant 2/\gamma$ for $0 \leqslant i < n$, $\left|(A^{-1})_{ij}\right| < 2/\gamma^2$ for $i \neq j$, and H is in minimal form. Empirically, taking α in the definition $\gamma = \alpha n$ to be as small as 2 is usually enough to ensure that these conditions hold with high probability. The private key is A, and the public key is $(v, d) \in \mathbb{Z}^n$. Since $v_i < d$ from the definition of the HNF (see Definition 3), and $d = O(\gamma^n)$ by virtue of the Hadamard bound on the size of the determinant of a matrix [8], it follows that the public key has size $O(n^2 \lg \gamma)$ or simply $O(n^2 \lg n)$ bits, while the private key, which is essentially P, has size n^2 bits.

- Encrypt: Let $m \in \{0, 1\}^{n-k}$ be the plaintext. Select a random subset $S \subset \{1 \ldots n\}$ with k elements. The encoding of m is a vector $r \in \mathbb{Z}^n$ such that $r_i = h$ for $i \in S$, otherwise $r_i \xleftarrow{\$} \{1 \ldots \sigma/2\}$ if $m_j = 0$, and $r_i \xleftarrow{\$} \{\sigma/2 + 1 \ldots \sigma\}$ if $m_j = 1$, where i corresponds to the j-th index not in S. Compute $r - \lfloor rH^{-1} \rfloor H$, which, because of the particularly simple structure of the minimal HNF (see Definition 4), has the form $(0, \ldots, 0, c)$. The cryptogram is $c \in \mathbb{Z}$, the only nonzero coefficient thereof.

- Decrypt: Let $c \in \mathbb{Z}$ be the ciphertext. Compute $c' \leftarrow (0, \ldots, 0, c)A^{-1} \in \mathbb{Q}^n$, which means simply $c' \leftarrow cA^{-1}_{(n-1)}$, and let $r' \leftarrow (c' - \lfloor c' \rceil)A$. Compute the error vector $e \in \{0, 1\}^n$ by letting $e_i \leftarrow 1$ whenever $r'_i < 0$, otherwise $e_i \leftarrow 0$, for all $0 \leqslant i < n$. Compute the recovered message encoding as $r \leftarrow r' + eA$. Let $S := \{i \mid r_i = h\}$ (this is the same set S chosen during encryption). For all $0 \leqslant i < n$ such that $i \notin S$, extract $m_j \leftarrow 0$ if $0 < r_i \leqslant \sigma/2$, and $m_j \leftarrow 1$ if $\sigma/2 < r_i \leqslant \sigma$, where i corresponds to the j-th index not in S.

Notice that, strictly speaking, this is only a trapdoor one way function, not a full semantically secure encryption scheme. To attain semantic security, a suitable transform like Fujisaki-Okamoto [7] should be used.

4. Improvements

The usual technique adopted to reduce space requirements and bandwidth occupation in lattice-based cryptosystems is to resort to certain structured matrices that correspond to ideals in polynomial rings [15, 17, 18].

The most popular choices are circulant matrices, associated to the polynomial ring $\mathbb{Z}[x]/(x^n - 1)$, and negacyclic matrices, which correspond to the polynomial ring $\mathbb{Z}[x]/(x^n + 1)$. Due to security concerns with the idea of working on a ring (where not all nonzero elements have inverses), Bernstein [3] suggests adopting a number field instead, specifically a field of form $\mathbb{Z}[x]/(x^n - x - 1)$ because of the very simple form of the irreducible polynomial $x^n - x - 1$, which yields nearly circulant matrices and fairly efficient arithmetic. More generally, one could consider the $n \times n$ matrices whose i-th row contains the coefficients of $a(x)x^i \bmod p(x)$ for some $a(x)$ and a fixed but arbitrary monic polynomial $p(x)$ of degree n without multiple roots (and preferably small coefficients). Such matrices correspond to the ideals of a polynomial ring $\mathbb{Z}[x]/p(x)$.

Unfortunately, this technique does not seem to improve the space requirements of GGH, nor, for that matter, those of GGH-YK-M. This is because the HNF is usually *not* in the same (structured) ring as the original matrix. Thus, for instance, HNF(A) in general is *not* circulant or negacyclic even though A displays such symmetries (except if A is a scalar matrix). Therefore, by resorting to circulant or similarly structured matrices one would apparently be able at most to reduce the size of private keys from n^2 down to n bits, but not that of public keys, which stay at $O(n^2 \lg \gamma)$ bits.

Contrary to this intuitive observation, one can still benefit from an underlying structure in the private key to reduce the size of the public key in a nontrivial way. This was first indicated by Smart and Vercauteren [23], but it seems to require computing the HNF of the lattice basis. Gentry and Halevi [9, Lemma 1] offer a proof of this property that avoids computing the HNF for the case $p(x) = x^n + 1$ (where n is a power of 2). We show that, in fact, it holds for any ideal matrix, regardless of the choice of $p(x)$, even though some choices may be more efficient (and possibly more secure) than others.

If matrix P in the Keygen algorithm is associated to a polynomial ring $\mathbb{Z}[x]/p(x)$, then matrix A is associated to a polynomial in the same ring, and although $H := \text{HNF}(A)$ does not display the ring symmetry (i.e. H is not circulant, etc), its rows still correspond to elements of that ring. Thus, if $a(x)$ is the polynomial associated to any row of H, then $xa(x) \bmod p(x)$ and $x^{-1}a(x) \bmod p(x)$ are two other (independent) vectors on the same lattice.

Given that $H_{(n-2)} = (0, \ldots, 0, 1, u)$ for some $u \in \mathbb{Z}$ (because H is assumed to be minimal), the polynomial associated to it is $ux^{n-1} + x^{n-2} = (ux + 1)x^{n-2}$, and hence $(ux + 1)x^i = (x^{-1})^{i-(n-2)}(ux + 1)x^{n-2}$ stands for yet another vector on that lattice for every $0 \leqslant i < n - 1$. Collecting all of these vectors together with $H_{(n-1)}$, one gets

$$H' = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ 0 & 0 & 0 & \ldots & 0 & 0 & d \end{bmatrix}, \tag{1}$$

which is an alternative basis for the same lattice, since all of its rows are linearly independent vectors from that lattice, and H' shares the same determinant d as H (and A). But because the HNF is unique, it also follows that $\text{HNF}(H') = H$, and by applying a straightforward Gaussian elimination on H', namely by changing $H'_{(n-1-j)} \leftarrow H'_{(n-1-j)} - uH'_{(n-j)}$

successively for $2 \leqslant j < n$ and then reducing modulo d, one gets

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & -(-u)^{n-1} \bmod d \\ 0 & 1 & \dots & 0 & 0 & -(-u)^{n-2} \bmod d \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & -u^2 \bmod d \\ 0 & 0 & \dots & 0 & 1 & u \\ 0 & 0 & \dots & 0 & 0 & d \end{bmatrix}, \tag{2}$$

and by comparing the result with the definition of minimal H (see Definition 4) yields $v_i = -(-u)^{n-1-i} \bmod d$ for $0 \leqslant i < n-1$.

Therefore, H (and its inverse) can be efficiently represented simply by $(u, d) \in \mathbb{Z}^2$. Because $0 < u < d$ and d satisfies the Hadamard bound for A, which is $d < \gamma^n$, it follows that H can be represented with only $2n \lg \gamma$ and hence $O(n \lg n)$ bits, a vast improvement over the naive $O(n^2 \lg \gamma)$ or $O(n^2 \lg n)$ size of the whole (v, d) for practical values of n (typically in the hundreds).

The remaining tasks are computing d and u from A. We now address these tasks individually. Our approach avoids both the computation of resultants and complex modifications of the extended Euclidean algorithm.

Computing the determinant d is accomplished by diagonalizing the projections of A onto a number of finite fields $\mathbb{F}_{q_0}, \dots, \mathbb{F}_{q_t-1}$ such that $d < \prod_k q_k$, since this enables computing $d \bmod q_k$ for each q_k, and then recovering d by means of the Chinese remainder theorem. This is possible as long as the polynomial $p(x)$ splits completely into n distinct linear factors over each of those fields. If that is the case, let $V \in \mathbb{F}_{q_k}^{n \times n}$ be the Vandermonde matrix built from the n distinct roots of $p(x)$ over $\mathbb{F}_{q_k}$, i.e. $V_{ij} := z_j^i$ with $p(z_j) = 0$ and $z_j \in \mathbb{F}_{q_k}$. Then V is invertible, and the diagonal form of A is $V^{-1}AV$ (the eigenvalues themselves are just the sequence of components of $A_{(0)}V$).

The obstacle to this approach is finding the fields $\mathbb{F}_{q_k}$ such that $p(x)$ splits in the required form over all of them. Exhaustive search via the factorization of an arbitrary $p(x)$ over candidate fields is far too expensive, even for fairly small n. One could reverse the reasoning and choose the roots of $p(x)$ first, but this only enables the computation of a single field $\mathbb{F}_q$ over which $p(x)$ splits, and because the coefficients of such a $p(x)$ are expected to be rather large, any private basis is usually very large as well, yielding an even larger determinant d which is likely to exceed q by a factor exponentially large in n, and hence precluding the recovery of d from its value mod q alone.

However, the circulant and negacyclic cases offer a much better prospect, since all that is required for $p(x)$ to split over $\mathbb{F}_{q_k}$ is that $n \mid q-1$ in the former case, and $2n \mid q-1$ in the latter. When n is a power of 2, the computation of the diagonal form of A amounts to a fast Fourier transform (more precisely, a fast number theoretic transform), which takes time $O(n \lg n)$ products by certain fixed roots of unity in $\mathbb{F}_{q_k}$. However, computation of the eigenvalues is fairly efficient even for general n, and as we shall see this extra flexibility in the choice of n tends, a bit surprisingly, to offer better key generation performance.

Assuming that the fields $\mathbb{F}_{q_k}$ are available and that the determinant d has been computed, the value of u, if it exists, can be computed as follows. The first row of H is expected to have the form $(1, u, 0, \dots, 0)$, associated to the polynomial $ux + 1$ in the

underlying polynomial ring. The rows of the matrix H^* corresponding to this polynomial spell the coefficients of $(ux + 1)x^j \bmod p(x)$. Thus H^* differs from H only in its last row, and it defines a sub-lattice of the lattice defined by H or, equivalently, by A.

Therefore, there must exist a matrix $M \in \mathbb{Z}^{n\times n}$ (actually in the same ring as A and H^*) such that $MA = H^*$. Let $A^\dagger$ be the classical adjoint (or adjugate) of A, i.e. $AA^\dagger = dI$. Then $dM = H^*A^\dagger$, and the peculiar structure of H^* reduces this to the Diophantine equation $dM_j - A^\dagger_{j-1}u = A^\dagger_j$ for all j. Thus, if any solution to this equation exists, it is $u = -A^\dagger_j/A^\dagger_{j-1 \pmod n} \bmod d$ for any j, which requires all $A^\dagger_j$ to be invertible mod d. However, this in turn actually requires only that $A^\dagger_0$ and $A^\dagger_1$ be invertible mod d, since then $A^\dagger_j = A^\dagger_0(A^\dagger_1/A^\dagger_0)^j = A^\dagger_0(-u)^j \bmod d$ as one can check by induction.

This provides a simple algorithm to determine at once whether HNF(A) is minimal, and if so, what the value of u in Equation 1 is. Indeed, $A^\dagger$ can be computed via the Chinese remainder theorem from $A^\dagger = dA^{-1} \bmod q_k$, and the extended Euclidean algorithm then yields $u \leftarrow -A^\dagger_1/A^\dagger_0 \bmod d$ or proves that no such u exists.

The efficient key pair generation this process enables, without a full HNF algorithm, arguably outweighs the practical restriction for $p(x) = x^n \pm 1$. This method works for any choice of n. Processing times are much smaller for this compact representation than they are for unstructured matrices. We report on experimental results in Section 6.

5. Security considerations

Adopting a structured matrix as the private key must be made carefully to avoid introducing weaknesses. The particular case $p(x) = x^n + 1$ where n is a power of 2 has received a considerable amount of attention in the literature. We now analyze how circulant lattices, corresponding to $p(x) = x^n - 1$, have the drawback of leaking a small amount of information on the private key, specifically $O(\lg n)$ bits thereof. As always, our analysis does not require n to be a power of 2. Admittedly, the security level attainable when generalizing n is less clear, though it seems unlikely that this would introduce any weakness that is not already present in the more extensively analyzed NTRU scenario, where prime n is the usual choice.

We begin by noticing that the sum $\lambda := \sum_j A_j$ is bound between $\gamma - n$ (when P is the all-one ring element) and γ (when P is zero). Let $\Lambda := \sum_j A^\dagger_j$. The following property holds:

Lemma 1. $d = \lambda\Lambda$.

Proof. By definition of adjugate matrix, $AA^\dagger = dI$. Then $A_{(j)}A^\dagger = dI_{(j)}$ and hence $\sum_j A_{(j)}A^\dagger = \sum_j dI_{(j)}$, which yields $(\lambda, \ldots, \lambda)A^\dagger = (d, \ldots, d)$, since the elements on each column of A are the same except for a circular permutation, and thus all columns of $\sum_j A_{(j)}$ take the value $\sum_j A_j = \lambda$. Now $(\lambda, \ldots, \lambda)A^\dagger = \lambda(1, \ldots, 1)A^\dagger$, which is simply $(\lambda\Lambda, \ldots, \lambda\Lambda)$ because $(1, \ldots, 1)A^\dagger = (\sum_j A^\dagger_j, \ldots, \sum_j A^\dagger_j) = (\Lambda, \ldots, \Lambda)$. Therefore $(\lambda\Lambda, \ldots, \lambda\Lambda) = (d, \ldots, d)$ which repeats the claim n times, i.e. $d = \lambda\Lambda$. □

Lemma 2. $(-u)^n - 1 \equiv 0 \bmod d$.

Proof. We show that $(0, \ldots, -(-u)^n + 1)$ is a lattice vector in the subspace generated by $H_{(n-1)} = (0, \ldots, d)$. Consider the lattice generated by

$$C^{(0)} = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & u & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ u & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix},$$

where the superscript denotes a stage in the Gaussian elimination process described below, with (0) indicating the original matrix. This is a sublattice of the original lattice, since it only involves rotations of the first row of H' defined by Equation 1. Applying Gaussian elimination to the last row as $C^{(j+1)}_{(n-1)} \leftarrow C^{(j)}_{(n-1)} + (-u)^{j+1} C_{(j)}$ for $j = 0, \ldots, n-1$, we get

$$C^{(n)} = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & u & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ 0 & 0 & 0 & \ldots & 0 & 0 & -(-u)^n + 1 \end{bmatrix}.$$

Thus $C^{(n)}_{(n-1)}$ is in the subspace spanned by $H'_{(n-1)}$, i.e. $C^{(n)}_{(n-1)} = \kappa H'_{(n-1)}$ for some κ. Thus $-(-u)^n + 1 = \kappa d$, i.e. $(-u)^n - 1 \equiv 0 \bmod d$ as claimed. □

Let $Z := \sum_j (-u)^j \bmod d$. Given that $A^\dagger_j = A^\dagger_0 (-u)^j \bmod d$, it follows that $\sum_j A^\dagger_j = A^\dagger_0 \sum_j (-u)^j \bmod d$ and thus $\Lambda = A^\dagger_0 Z \bmod d$. At first glance this equation might seem to provide a means to recover the full $A^\dagger_0$ by inverting $Z \bmod d$. That this cannot actually happen is established by the following property:

Lemma 3. $\Lambda \mid \gcd(Z, d)$*, and hence* Z *is not invertible* mod d.

Proof. From $A^\dagger_0 Z = \Lambda \bmod d$ and Lemma 2 it follows that $\lambda A^\dagger_0 Z = \lambda\Lambda = 0 \bmod d$ and since, by the key generation requirement of Section 4, $A^\dagger_0$ itself is invertible mod d, then $\lambda Z = 0 \bmod d$, i.e. $\lambda Z = Z' d = Z' \lambda \Lambda$ for some integer Z', meaning that $Z = Z'\Lambda$, i.e. Z itself is a multiple of Λ, and hence cannot be invertible mod d by virtue of having the common factor Λ with d. □

However, equation $\Lambda = A^\dagger_0 Z \bmod d$ does reveal a small piece of information on $A^\dagger_0$. Indeed, $\Lambda = A^\dagger_0 Z + \kappa d = A^\dagger_0 Z' \Lambda + \kappa\lambda\Lambda$ for some κ, and hence $1 = A^\dagger_0 Z' + \kappa\lambda$ by removing the common factor Λ, or simply $1 = A^\dagger_0 Z' \bmod \lambda$. This reveals $A^\dagger_0 \bmod \lambda = Z'^{-1} \bmod \lambda$

as long as Z' is invertible mod λ. However, this amounts to revealing only $O(\lg n)$ bits of the private value $A_0^\dagger$.

On the constructive side, $(u+1)\Lambda = A_0^\dagger(u+1)\sum_j(-u)^j \bmod d = -A_0^\dagger((-u)^n - 1) \bmod d = 0$, $(u+1)\Lambda = \xi\lambda\Lambda$ for some ξ, and hence $\lambda \mid u+1$. Thus λ is a common factor between d and $u+1$, and can be factored out by publishing the public key as the triple $(d/\lambda, (u+1)/\lambda, \lambda)$ instead of the pair (u, d), saving $O(\lg n)$ bits.

This also shows that the attack cannot be extended to recover the whole matrix $A \bmod \lambda$ (from which A could be extracted immediately) from $A^\dagger \bmod \lambda$. Because $u+1 = 0 \bmod \lambda$ and hence $-u = 1 \bmod \lambda$, it follows that $A_j^\dagger \bmod = A_0^\dagger(-u)^j \bmod \lambda$ (this equality holds because $\lambda \mid d$) and hence $A_j^\dagger = A_0^\dagger \bmod \lambda$ for all j, so that $A^\dagger = A_0^\dagger U \bmod \lambda$ where U is the (singular) all-one matrix. Therefore the adjugate mapping mod λ cannot be inverted to recover A from $A^\dagger \bmod \lambda$.

Interestingly, this attack does not apply to negacyclic lattices (or, for that matter, most or perhaps all other ideal lattices), because Lemma 1 does not hold, i.e. the determinant, in general, is not the product of a linear combination of the components of A and a linear combination of the components of $A^\dagger$.

6. Experimental results

We implemented the improved encryption scheme in Java running on an Intel i5-3210M 2.5 GHz platform under 64-bit Windows 7.

To facilitate comparison with the literature [6], where timings, obtained from an implementation in C/C++, are only available on an AMD 1.6 GHz platform, our speeds are shown scaled down by a factor 1.6/2.5 on Table 1. Performance turned out to be already highly competitive with the prior state of the art, in spite of the adoption of Java rather than C/C++.

We disregard lattice dimensions smaller than 350, since they are susceptible to attacks [6], and we set n to be either a prime or a power of 2. We provide data for dimensions around 512 as well, going somewhat beyond the dimensions found in that reference. The times needed to gather suitable primes for the Chinese remainder theorem are not included since they are precomputed only once and stored. For simplicity, we only consider the circulant ring $\mathbb{Z}[x]/(x^n - 1)$, since the times corresponding to negacyclic ring $\mathbb{Z}[x]/(x^n + 1)$ would be very similar to those corresponding to the circulant case.

Table 1. Timings (in seconds)

source	(n, σ, h, k)	keygen (s)	encrypt (ms)	decrypt (ms)
[6]	(350, 256, 526, 64)	1662.55	60.0	170.0
ours	(353, 256, 526, 64)	0.48	0.7	88.5
[6]	(400, 256, 601, 64)	3127.17	70.0	270.0
ours	(401, 256, 601, 64)	0.65	0.8	132.3
ours	(509, 256, 769, 80)	1.30	1.2	448.0
ours	(512, 256, 769, 80)	4.30	1.3	278.8

By design, we only consider private keys whose HNF is minimal. To this end, we adopted a rejection sampling strategy, generating uniformly random private keys and

discarding those that do not satisfy the desired property, until finding one that does.

Interestingly, prime values of n tend to yield lattices with minimal HNF far more often than composite n. Empirically, the probability that a random circulant matrix A has a minimal HNF is heavily affected by the choice of lattice parameters, particularly its dimension n, being roughly $O(1/D)$ where D is the number of irreducible factors of $x^n - 1$. Tourloupis [24] addresses this issue (for a generic matrix A, not necessarily circulant) by sieving the randomly sampled A to have prime or near-prime determinant, thus ensuring that it has a 99% probability of sporting a minimal HNF. However, choosing n itself to be prime increases that probability to the same level (since the number of irreducible factors of $x^n - 1$ coincide with the number of factors of n), without having to resort to primality testing during key generation. This behavior is only counterbalanced for composite n when the FFT is available, in which case processing is fast enough to roughly compensate for the rejection sampling overhead.

Key sizes are essentially the same in our proposal for a given dimension n regardless of the choice of $p(x)$. Sample public key sizes are listed on Table 2.

Table 2. Public key sizes (in bits)

source	(n, σ, h, k)	$\lvert pk \rvert$
[6]	$(350, 256, 526, 64)$	1157800
ours	$(353, 256, 526, 64)$	6682
[6]	$(400, 256, 601, 64)$	1543200
ours	$(401, 256, 601, 64)$	7738
[6]†	$(512, 256, 769, 80)$	2621440
ours	$(512, 256, 769, 80)$	10240

† Inferred.

7. Conclusion

We have shown how to enhance the GGH-YK-M scheme by Barros and Schechter, reducing its public key size by an order of complexity from $O(n^2 \lg n)$ down to $O(n \lg n)$ bits. The bandwidth savings stem from the technique first put forward by Smart and Vercauteren technique, which we optimize in a simpler and more efficient way than the Gentry-Halevi method. As a result, key generation times decrease as compared to the Barros-Schechter variant by more than 3 orders of magnitude. Besides the key generation speedup, encryption becomes almost 2 orders of magnitude faster; decryption is about twice as fast though the reason for the improvement in this particular operation could be simply related to different implementation details. Our benchmarks were obtained using Java; a C/C++ implementation is likely to improve the timings even more.

An intriguing line for follow-up research is to assess the impact of extending the proposed method to lattices whose HNF is only near-minimal, say, having the two rightmost columns in nontrivial form. This is observed far more often than a minimal HNF when n is composite and might reduce key generation times considerably. The key and cryptogram sizes remain the same, because the sizes of the elements on each row of those nontrivial columns are bound by complementary factors of the determinant. While there are more factors to tackle for encryption and decryption, they are also smaller than in the minimal HNF case, so processing might end up being faster for those operations as well.

Our results show that the proposed techniques constitute a viable option to help minimize the cost of the GGH-YK-M scheme, and possibly for other lattice-based protocols, regarding both key size and processing times. We leave the application of the proposed techniques to somewhat (levelled) homomorphic encryption as a further research problem.

References

[1] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*, volume 9. Society for Industrial and Applied Mathematics (SIAM), 1994.

[2] D. J. Bernstein, J. Buchmann, and E. Dahmen. *Post-Quantum Cryptography*. Springer, Heidelberg, Deutschland, 2008.

[3] Daniel J. Bernstein. A subfield-logarithm attack against ideal lattices. Blog entry, February 2014. `http://blog.cr.yp.to/20140213-ideal.html`.

[4] Massimo Chenal and Qiang Tang. On key recovery attacks against existing somewhat homomorphic encryption schemes. In *International Conference on Cryptology and Information Security in Latin America – Latincrypt 2014*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[5] Henri Cohen. *A course in computational algebraic number theory*, volume 138. Springer, 1993.

[6] Charles F. de Barros and L. Menasché Schechter. GGH may not be dead after all. In *XXXV Congresso Nacional de Matemática Aplicada e Computacional – CNMAC 2014*. Sociedade Brasileira de Matemática Aplicada e Computacional – SBMAC, 2014.

[7] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – Crypto 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, USA, 1999. Springer.

[8] D. J. H. Garling. *Inequalities: A Journey into Linear Analysis*. Cambridge, 2007.

[9] Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology – Eurocrypt 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.

[10] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology – CRYPTO '97*, pages 112–131. Springer, 1997.

[11] J. Hoffstein, J. Pipher, and J. Silverman. NTRU: A ring-based public key cryptosystem. In J. P. Buhler, editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, Oregon, USA, 1998.

[12] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[13] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339, San Francisco, CA, USA, 2011. Springer.

[14] Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On CCA-secure somewhat homomorphic encryption. In *International Conference on Selected Areas in Cryptography – SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2012.

[15] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.

[16] Robert J McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978.

[17] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 356–365. IEEE, 2002.

[18] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[19] Phong Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from CRYPTO '97. In *Advances in Cryptology – CRYPTO '99*, pages 288–304. Springer, 1999.

[20] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[21] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, 53(2):201–224, 1987.

[22] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1995.

[23] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.

[24] Vasilios Evangelos Tourloupis. Hermite normal forms and its cryptographic applications. Master's thesis, University of Wollongong, 2013.

[25] M. Yoshino and N. Kunihiro. Improving GGH cryptosystem for large error vector. In *International Symposium on Information Theory and its Applications – ISITA 2012*, pages 416–420. IEEE, 2012.

Expanding a Lattice-based HVE Scheme

Karina Mochetti[1], **Ricardo Dahab**[1]

[1]Instituto de Computação (UNICAMP)
Av. Albert Einstein, 1251, 13083-852, Campinas-SP, Brazil

{mochetti,rdahab}@ic.unicamp.br

Abstract. *Functional encryption systems provide finer access to encrypted data by allowing users to learn functions of encrypted data. A Hidden-Vector Encryption Scheme (HVE) is a functional encryption primitive in which the ciphertext is associated with a binary vector $\boldsymbol{w}$ and the secret key is associated with a special binary vector $\boldsymbol{v}$ that allows "don't care" entries. The decryption is only possible if the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are the same for all elements, except the "don't care" entries in $\boldsymbol{v}$. HVE schemes are used to construct more sophisticated schemes that support conjunctive and range searches. In this work we show how to expand the basic fuzzy IBE scheme of Agrawal et al. (PKC 2012) to a hierarchical HVE scheme. We also show how the version using ideal lattices affects the security proof.*

1. Introduction

In a functional encryption system, secret keys allow users to learn functions of encrypted data, i.e., for a message m and a value k it is possible to evaluate a function $f(k, m)$ given the encryption of m and a secret key sk_k, thus providing a much finer control of decryption capabilities. Some functional encryption primitives are Identity-Based Encryption (IBE), Attribute-Based Encryption (ABE), Inner-Product Encryption (IPE) and Hidden-Vector Encryption (HVE) [Boneh et al. 2011] .

In a Hidden-Vector Encryption (HVE) scheme the ciphertext is associated with a binary vector $\boldsymbol{w}$ and the secret key is associated with a special binary vector $\boldsymbol{v}$ that allows "don't care" entries (denoted by $\star$). The decryption is only possible if the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are the same for all elements that are not represented by $\star$ in vector $\boldsymbol{v}$.

HVE schemes are used on more sophisticated functional encryption schemes that support conjunctive and range searches, for example. The first scheme was proposed in [Boneh and Waters 2007] based on bilinear groups and proved secure under the selective model. Other schemes, also based on bilinear groups, were developed, such as the scheme proposed in [Iovino and Persiano 2008] that uses bilinear groups of prime order and the scheme presented in [Caro et al. 2011] that is the first fully secure construction known.

Most lattice-based HVE schemes known can be built from Inner Product Encryption, such as [Agrawal et al. 2011] and [Abdalla et al. 2012]. Such construction was first presented in [Katz et al. 2008]. Also, the basic fuzzy IBE scheme presented in [Agrawal et al. 2012] can clearly be seen as an HVE scheme. The main focus of this work is to expand the underlying scheme to an ideal lattice-based HVE scheme and to a hierarchical HVE scheme.

Ideal lattices are a generalization of cyclic lattices, first presented in [Micciancio 2002], in which the lattice corresponds to ideals in a ring $\mathbb{Z}[x]/\langle f(x)\rangle$, for some irreducible polynomial function $f(x)$. They can be used to decrease the parameters needed to describe a lattice and its basis pattern can be used to improve the matrix multiplication complexity.

For cryptosystems that use a Trusted Third Part, as HVE schemes, it is convenient to have a hierarchy of certificate authorities, that is, the root certificate authority can issue certificates for other certificate authorities, which can issue certificates for users. A scheme in which a user in level t can use his/her secret key to derive a secret key for a user at level $t+1$ is called hierarchical, as introduced in [Hanaoka et al. 2009]. This reduces the workload on a Thrusted Third Part as it does not need to generate all public and master keys.

Our Contributions. In this work we show how to expand the basic fuzzy IBE scheme of [Agrawal et al. 2012] to a hierarchical HVE scheme. We show that this expansion has the same security, based on the Learning With Errors Problem (LWE), as the original scheme, while having the new feature which reduces the role of the Thrusted Third Part in the scheme. We also show a version using ideal lattices, that has the advantages of smaller key sizes and more efficient matrix multiplication, and how it affects the security proof.

2. Definitions

For any integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q and we represent $\mathbb{Z}_q$ as integers in $(q/2, q/2]$. We let $\mathbb{Z}_q^{n\times m}$ denote the set of $n \times m$ matrices with entries in $\mathbb{Z}_q$. We use capital letters (e.g. $\boldsymbol{A}$) to denote matrices, bold lowercase letters (e.g. $\boldsymbol{w}$) to denote vectors. The notation $\boldsymbol{A}^\top$ denotes the transpose of matrix $\boldsymbol{A}$. When we say a matrix defined over $\mathbb{Z}_q$ has *full rank*, we mean that it has full rank modulo each prime factor of q. If $\boldsymbol{A}_1$ is an $n \times m$ matrix and $\boldsymbol{A}_2$ is an $n \times m'$ matrix, then $[\boldsymbol{A}_1|\boldsymbol{A}_2]$ denotes the $n \times (m+m')$ matrix formed by concatenating the columns of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. If $\boldsymbol{w}_1$ is a length-m vector and $\boldsymbol{w}_2$ is a length m' vector, then we let $[\boldsymbol{w}_1|\boldsymbol{w}_2]$ denote the length-$(m+m')$ vector formed by concatenating $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. However, when doing matrix-vector multiplication we always view vectors as column vectors. For a vector $\boldsymbol{v}$ we define $|\boldsymbol{v}| = \sqrt{\sum \boldsymbol{x}_i^2}$ as the norm of vector $\boldsymbol{v}$, and for matrix $\boldsymbol{A}$, we define $|\boldsymbol{A}| = \max|\boldsymbol{A}\boldsymbol{x}|$, for $|\boldsymbol{x}| = 1$, as the norm of matrix $\boldsymbol{A}$. We say a function $f(n)$ is *negligible* if it is $O(n^{-c})$ for all $c > 0$, and we use $\mathrm{negl}(n)$ to denote a negligible function of n. We say that function $f(n)$ is *polynomial* if it is $O(n^c)$ for some $c > 0$, and we use $\mathrm{poly}(n)$ to denote a polynomial function of n. We say an event occurs with *overwhelming probability* if its probability is $1 - \mathrm{negl}(n)$. Given a polynomial $f(x)$ we say a ring $\mathbb{Z}[x]/\langle f(x)\rangle$ is the set of all polynomials $g(x) \mod f(x)$ with coefficients in $\mathbb{Z}$. The notation $g(x) \otimes h(x)$ denotes the multiplication of polynomials $g(x)$ and $h(x) \in \mathbb{Z}[x]/\langle f(x)\rangle$ modulo $f(x)$. The notation $[d]$ denotes the set of positive integers $\{1, 2, \ldots, d\}$.

2.1. Hidden Vector Encryption

Based on the definition of predicate encryption by [Katz et al. 2008], we have that a Hidden Vector Encryption Scheme consists of the following four algorithms:

Setup(1^n). Takes as input security parameter λ and outputs public-key mpk and master secret key msk.

KeyGen(mpk, msk, v). Takes as input public-key mpk, master secret key msk and a vector $\boldsymbol{v} \in \{0, 1, \star\}^l$ and outputs a secret key sk.

Enc($mpk, \mathfrak{m}, w$). Takes as input public parameters, message $\mathfrak{m}$ from some associated message space, public-key mpk, a vector $\boldsymbol{w} \in \{0, 1\}^l$ and outputs a ciphertext C.

Dec(mpk, sk, C). Takes as input public-key mpk, ciphertext C, secret key sk and outputs the message $\mathfrak{m}$.

Suppose ciphertext C is obtained by running Enc on input mpk, message $\mathfrak{m}$ and vector $\boldsymbol{w}$ and that sk is a secret key obtained through a call of KeyGen using the same mpk and vector $\boldsymbol{v}$. Then Dec, on input mpk, C and sk returns $\mathfrak{m}$, except with negligible probability, if and only if $v_i = w_i$, for all $i \in [l]$ such that $v_i \neq \star$.

For a hierarchical scheme, the KeyGen algorithm is replaced by the Derive algorithm:

Derive(mpk, sk_{t-1}, v). Takes as input public-key mpk, secret key sk_{t-1} for hierarchical level $t-1$ and a vector $\boldsymbol{v} \in \{0, 1, \star\}^l$, and outputs a secret key sk_t for level t.

Security is modelled by means of a game between a challenger $\mathcal{B}$ and a probabilistic polynomial-time adversary $\mathcal{A}$. In this work, we achieve *selective atribute* security, meaning that $\mathcal{A}$ must declare its *challenge vectors* before seeing the public-key.

Init. $\mathcal{A}$ outputs challenge vectors $\boldsymbol{w}_0^\star, \boldsymbol{w}_1^\star$.

Setup. The challenger $\mathcal{B}$ runs the Setup algorithm to generate public-key mpk which it gives to the adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ is given oracle access to KeyGen$(mpk, msk, \cdot)$.

Challenge. The adversary $\mathcal{A}$ gives a pair of messages $(\mathfrak{m}_0, \mathfrak{m}_1)$ to the challenger $\mathcal{B}$. Then $\mathcal{B}$ chooses random $\eta \xleftarrow{\$} \{0, 1\}$, encrypts $\mathfrak{m}_\eta$ under $\boldsymbol{w}_\eta$ and sends the resulting ciphertext to $\mathcal{A}$.

Phase 2. The same as Phase 1.

Guess. The challenger $\mathcal{A}$ must output a guess η' for η.

If the advantage of every probabilistic polynomial time adversary $\mathcal{A}$ is defined to be $|\Pr[\eta' = \eta] - \frac{1}{2}|$, then we say the scheme has indistinguishability under chosen-plaintext attack under the selective model (IND-sAT-CPA for short).

2.2. Lattices

This section presents the collection of results from [Agrawal et al. 2011, Lyubashevsky et al. 2010, Micciancio and Regev 2004, Cash et al. 2010] that we will need for our construction and proof of security.

An m-dimensional lattice Λ is a full-rank discrete subgroup of $\mathbb{R}^m$. A basis of Λ is a linearly independent set of vectors whose span is Λ. We will focus on integer lattices and among these we will focus on the q-ary lattices defined as follows: for any integer

$q \geq 2$ and any $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, we define

$$\Lambda_q^{\perp}(\boldsymbol{A}) := \{\boldsymbol{e} \in \mathbb{Z}^m : \boldsymbol{A}\boldsymbol{e} = 0 \mod q\}$$
$$\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) := \{\boldsymbol{e} \in \mathbb{Z}^m : \boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \mod q\}$$
$$\Lambda_q(\boldsymbol{A}) := \{\boldsymbol{e} \in \mathbb{Z}^m : \exists\, \boldsymbol{s} \in \mathbb{Z}_q^m \text{ with } \boldsymbol{A}^{\top}\boldsymbol{s} = \boldsymbol{e} \mod q\}.$$

2.2.1. Ideal Lattices

Let $\mathcal{I}$ be an ideal of the ring $\mathcal{R} = \mathbb{Z}[x]/\langle f(x)\rangle$, i.e., a subset of $\mathcal{R}$ that is closed under addition and multiplication. The ideal $\mathcal{I}$ is a sublattice of $\mathbb{Z}^n$. For a ring $\mathcal{R} = \mathbb{Z}[x]/\langle f(x)\rangle$ we can define the basis of the ideal lattice $\Lambda_q^{\perp}(\boldsymbol{A})$, with $\boldsymbol{A} = \mathsf{rot}_f(g) \in \mathbb{Z}_q^{n \times n}$, where each row i of $\boldsymbol{A}$ is given by the coefficients of $x^i g(x) \mod f(x)$ for $i \in \{0, n-1\}$.

For a polynomial $g(x) \in \mathcal{R}$, we can represent it as a vector $\boldsymbol{a}$ where for each $i \in \{0, n-1\}$, a_i is the coefficient of x^i in $g(x)$. We assume that any polynomial is a vector, and $\boldsymbol{a} \otimes \boldsymbol{b}$ is the multiplication of the polynomials represented by vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. For a ring $\mathcal{R}$, we have that $\hat{\boldsymbol{g}} \in \mathcal{R}^k$ is a vector of k polynomials in $\mathcal{R}$. Since polynomials are easily represented as vectors, we denote by $\hat{\boldsymbol{v}}$ any concatenation of vectors, i.e., $\hat{\boldsymbol{v}} = [\boldsymbol{v}_0|\ldots|\boldsymbol{v}_k]$, with $\boldsymbol{v}_i$ a vector.

Note that if $f(x) = x^n + 1$, then the matrix $\boldsymbol{A} = \mathsf{rot}_f(g) \in \mathbb{Z}_q^{n \times n}$ is an anti-circulant matrix and for $f(x) = x^n - 1$, we have that the matrix $\boldsymbol{A} = \mathsf{rot}_f(g) \in \mathbb{Z}_q^{n \times n}$ is a circulant matrix. These lattices are called cyclic lattices and they are a special class of ideal lattices. The two main advantages of using ideal lattices are: The basis matrix $n \times m$ can be built from a polynomial with degree m, which results in smaller key sizes. The multiplication of a matrix that is a basis for an ideal lattice by a vector can be done in an efficient way [Pan 2001].

For simplicity, we define the $\mathsf{Rot}()$ function, that takes a vector $\hat{\boldsymbol{a}} \in \mathcal{R}^k$ and also expands it to a matrix as follows:

$$\mathsf{Rot}(\hat{\boldsymbol{a}}) = \left[\ \mathsf{rot}_f(\boldsymbol{a}_0)|\mathsf{rot}_f(\boldsymbol{a}_1)|\ldots|\mathsf{rot}_f(\boldsymbol{a}_{k-1})\ \right]$$

2.2.2. Sampling Algorithms

This section gives the main definitions and theorems over lattices used to generate trapdoor functions.

Definition 1. *Let $\boldsymbol{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k\}$ be a set of vectors in $\mathbb{R}^m$. Let $|\boldsymbol{S}|$ denotes the length of the longest vector in $\boldsymbol{S}$, i.e., $\max_{1 \leq i \leq k} |\boldsymbol{s}_i|$, and $\widetilde{\boldsymbol{S}} := \widetilde{\boldsymbol{s}}_1, \ldots, \widetilde{\boldsymbol{s}}_k \subset \mathbb{R}^m$ denotes the* Gram-Schmidt orthogonalization *of the vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k$. We refer to $|\widetilde{\boldsymbol{S}}|$ as the* Gram-Schmidt norm *of $\boldsymbol{S}$.*

Definition 2. *Let L be a discrete subset of $\mathbb{Z}^n$. For any vector $\boldsymbol{c} \in \mathbb{R}^n$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, let $\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{w}) := \exp\left(-\pi|x - \boldsymbol{c}|^2/\sigma^2\right)$ be the Gaussian function on $\mathbb{R}^n$ with center $\boldsymbol{c}$ and parameter σ. Let $\rho_{\sigma,\boldsymbol{c}}(L) := \sum_{\boldsymbol{w} \in L} \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{w})$ be the discrete integral of $\rho_{\sigma,\boldsymbol{c}}$ over L, and let $\mathcal{D}_{L,\sigma,\boldsymbol{c}}$ be the discrete Gaussian distribution over L with center $\boldsymbol{c}$ and parameter σ. Specifically, for all $\boldsymbol{v} \in L$, we have $\mathcal{D}_{L,\sigma,\boldsymbol{c}}(\boldsymbol{v}) = \frac{\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{v})}{\rho_{\sigma,\boldsymbol{c}}(L)}$. For notational*

convenience, $\rho_{\sigma,0}$ and $\mathcal{D}_{L,\sigma,0}$ are abbreviated as ρ_σ and $\mathcal{D}_{L,\sigma}$ respectively.

The following theorem shows how to sample an essentially uniform matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ along with a basis $\boldsymbol{S}$ of $\Lambda_q^\perp(\boldsymbol{A})$ with low Gram-Schmidt norm.

Theorem 1. *[Alwen and Peikert 2009] Let q, n, m be positive integers with $q \geq 2$ and $m \geq 6n\lg q$. There is a probabilistic polynomial-time algorithm $\mathsf{TrapGen}(q, n, m)$ that outputs a pair $(\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}, \boldsymbol{S} \in \mathbb{Z}^{m\times m})$ such that $\boldsymbol{A}$ is statistically close to uniform in $\mathbb{Z}_q^{n\times m}$ and $\boldsymbol{S}$ is a basis for $\Lambda_q^\perp(\boldsymbol{A})$, satisfying $|\widetilde{\boldsymbol{S}}| \leq O(\sqrt{n\log q})$ and $|\boldsymbol{S}| \leq O(n\log q)$ with overwhelming probability in n.*

The following theorem shows an adaptation of Ajtai's trapdoor key generation algorithm for ideal lattices.

Theorem 2. *[Stehlé et al. 2009] Let n, σ, q, k be positive integers with $q \equiv 3 \mod 8$, $k \geq \lceil \log q + 1 \rceil$, let n be a power of 2 and let $f(x) = x^n + 1$ be a degree n polynomial in $\mathbb{Z}[x]$. Then, there is a probabilistic polynomial-time algorithm $\mathsf{IdealTrapGen}(q, n, k, \sigma, f)$ that outputs a pair $(\vec{a} \in \mathcal{R}^k, \boldsymbol{S} \in \mathbb{Z}^{kn\times kn})$ such that $\vec{a}$ is statistically close to uniform in $\mathcal{R}^k$ and $\boldsymbol{S}$ is a basis for $\Lambda_q^\perp(\boldsymbol{A})$, for $\boldsymbol{A} = \mathsf{Rot}_f(\vec{a})$, satisfying $|\boldsymbol{S}| = O(n\log q\sqrt{\omega(\log n)})$ with overwhelming probability in n.*

The following theorems give a few sample algorithms used in lattice-based schemes.

Theorem 3. *[Gentry et al. 2008] Let $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ be a full rank matrix, let $\boldsymbol{S}$ be a short basis of $\Lambda_q^\perp(\boldsymbol{A})$ and let σ be the Gaussian parameters. For q, m, n integers such that $q > 2$ and $m > n$ and $\sigma > \|\boldsymbol{S}\|\cdot\omega(\sqrt{\log m})$, there is a probabilistic polynomial algorithm $\mathsf{SamplePre}(\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{u}, \sigma)$ that outputs a vector $\boldsymbol{e} \in \mathbb{Z}^m$ statistically close to $\mathcal{D}_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}),\sigma}$.*

Theorem 4. *[Gentry et al. 2008] Let $\sigma, \vec{c}$ be Gaussian parameter, let $\boldsymbol{A} \in \mathbb{Z}^{n\times m}$ be a matrix and let n, m be integers such that $\sigma \geq \|\boldsymbol{A}\| \cdot \omega(\sqrt{\log n})$. Then there is a probabilistic polynomial algorithm $\mathsf{SampleGaussian}(\boldsymbol{A}, \sigma)$ that outputs a vector $\boldsymbol{e} \in \mathbb{Z}^m$ statistically close to $\mathcal{D}_{\Lambda(\boldsymbol{A}),\sigma}$.*

Theorem 5. *[Agrawal et al. 2010] Let $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ be a full rank matrix, let $\boldsymbol{S}$ be a short basis of $\Lambda_q^\perp(\boldsymbol{A})$, let $\boldsymbol{B} \in \mathbb{Z}_q^{n\times m_1}$ be a matrix and let σ be a Gaussian parameter. For q, m, n be integers such that $q > 2$ and $m > 2n\log q$ and $\sigma > \|\boldsymbol{S}\| \cdot \omega(\sqrt{\log(m + m_1)})$, there is a probabilistic polynomial algorithm $\mathsf{SampleBasisLeft}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \sigma)$ that outputs a new basis $\boldsymbol{T} \in \mathbb{Z}^{n\times m+m_1}$ for lattice $\Lambda_q^\perp(\boldsymbol{F})$, with $\boldsymbol{F} = (\boldsymbol{A}|\boldsymbol{B})$.*

Theorem 6 ([Agrawal et al. 2010])**.** *Let $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{B} \in \mathbb{Z}_q^{n\times m_1}$ be full rank matrices, let $\boldsymbol{S}$ be a short basis for $\Lambda_q^\perp(\boldsymbol{B})$, let $\boldsymbol{R} \in \{-1, 1\}^{m\times m1}$ be a uniform random matrix and let σ be a Gaussian parameter. Let q, m, n be integers such that $q > 2$ and $m > n$ and let $\sigma > \|\boldsymbol{S}\| \cdot \sqrt{m} \cdot \omega(\sqrt{\log m})$. Then there is a probabilistic polynomial algorithm $\mathsf{SampleBasisRight}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{S}, \sigma)$ that outputs a new basis $\boldsymbol{T} \in \mathbb{Z}^{n\times m+m_1}$ for lattice $\Lambda_q^\perp(\boldsymbol{F})$, with $\boldsymbol{F} = (\boldsymbol{A}|\boldsymbol{A}\boldsymbol{R} + \boldsymbol{B})$.*

2.2.3. Learning With Errors Problem

The Learning With Errors problem, or LWE, is the problem of determining a secret vector over $\mathbb{F}_q$ given a polynomial number of noisy inner products. The decision variant is to distinguish such samples from random. More formally, we define the (average-case) problem as follows:

Definition 3. *[Regev 2005] Let $n \geq 1$ and $q \geq 2$ be integers, and let χ be a probability distribution on $\mathbb{Z}_q$. For $\boldsymbol{r} \in \mathbb{Z}_q^n$, let $A_{\boldsymbol{r},\chi}$ be the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a vector $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}_q$ according to χ, and outputting $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{r} \rangle + e)$. The* decision-$\mathsf{LWE}_{q,n,\chi}$ *problem is: for uniformly random $\boldsymbol{r} \in \mathbb{Z}_q^n$, given a $\mathrm{poly}(n)$ number of samples that are either (all) from $A_{\boldsymbol{r},\chi}$ or (all) uniformly random in $\mathbb{Z}_q^n \times Z_q$, output 0 if the former holds and 1 if the latter holds.*

The hardness of the LWE problem is summarized in the following

Definition 4. *For $\alpha \in (0,1)$ and an integer $q > 2$, let $\overline{\Psi}_\alpha$ denote the probability distribution over $\mathbb{Z}_q$ obtained by choosing $x \in \mathbb{R}$ according to the normal distribution with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ and outputting $\lfloor qx \rceil$.*

Theorem 7. *[Regev 2005] Let n, q be integers and $\alpha \in (0,1)$ such that $q = \mathrm{poly}(n)$ and $\alpha q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves* decision-$\mathsf{LWE}_{q,n,\overline{\Psi}_\alpha}$*, then there exists an efficient quantum algorithm that approximates* SIVP *and* GapSVP *to within $\widetilde{O}(n/\alpha)$ in the worst case.*

The ideal-LWE Problem is the same as described above, but with $\boldsymbol{a}$ chosen uniformly in $\mathbb{Z}_q[x]/f(x)$. The hardness of the ideal-LWE problem is summarized in the following:

Theorem 8 ([Lyubashevsky et al. 2010])**.** *Let n, q be integers and $\alpha > 0$ such that $q \geq 2$, $q = 1 \mod m$ and q be a $\mathrm{poly}(n)$-bounded prime such that $\alpha q \geq \omega(\sqrt{\log n})$. If there exists an efficient (possibly quantum) algorithm that solves* decision-ideal-$\mathsf{LWE}_{q,n,\Upsilon_\alpha}$*, then there exists an efficient quantum algorithm that solves γ-SIVP and γ-SVP for $\gamma = \widetilde{O}(n/\alpha)$ in the worst case.*

2.3. Shamir's Secret Sharing

Shamir's Secret Sharing [Shamir 1979] is a threshold scheme, i.e., a scheme to divide data into n parts in a way that it is only possible to recover the data with at least r parts, for $r \leq n$. The scheme is based on polynomial interpolation, i.e., for data d we create shares by choosing a random polynomial p of degree $r-1$ with $p(0) = d$, then each share piece d_i, for $i \in [n]$ will be a point defined by the polynomial, so $d_i = p(i)$

To recover the data, the polynomial is rebuilt using r points. Several algorithms for polynomial evaluation and interpolation are known and can be used. One of the most efficient method known is the Lagrange Algorithm, which calculates r polynomials $l_j(x)$, called Lagrangian coefficients, based on the r given points $(x_j, y_j) = (i, d_i)$ and reconstruct the polynomial $p(x)$ calculating

$$p(x) = \sum_{j=0}^{k} y_j l_j(x)$$

where,

$$l_j(x) = \prod_{m=0}^{k} \frac{x - x_m}{x_j - x_m}.$$

The data will, therefore, be $d = p(0) = \sum_{j=0}^{k} y_j l_j(0)$.

Lemma 1. *([Agrawal et al. 2012, Lemma 3]) Let $\beta = (l!)^2$. Given $k \leq l$ numbers $x_1, \cdots, x_k \in [1, l]$ define the lagrangian coefficients*

$$l_j = \prod_{i \neq j} \frac{-x_i}{x_j - x_i}.$$

Then, for every $1 \leq j \leq k$, the value βl_j is an integer, and $|\beta l_j| \leq \beta^2 \leq (l!)^4$.

3. Hierarchical Lattice-Based HVE Scheme

In this section we provide our hierarchical HVE scheme, with its correctness and security proof.

3.1. Our Construction

Let n be the security parameter, σ be the Gaussian parameter, l be the length of vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, r the threshold value and h the hierarchical depth.

$\mathsf{Setup}(1^n)$. On input of security parameter n, the algorithm generates the public and secret keys as follows:

(i) run the $\mathsf{TrapGen}(n, q, \sigma)$ algorithm to select uniformly l matrices $\boldsymbol{A}_i \in \mathbb{Z}^{n \times m}$ (for $i \in [l]$), with a short basis $\boldsymbol{T}_{\boldsymbol{A}_i} \in \mathbb{Z}^{m \times m}$ for $\Lambda_q^{\perp}(\boldsymbol{A}_i)$;
(ii) choose uniformly random vector $\boldsymbol{u} \in \mathbb{Z}_q^n$;
(iii) choose random matrices $\boldsymbol{A}_{i,b,j}$ and $\boldsymbol{B}_i$ (for $i \in [l]$, $b \in \{0, 1\}$ and $j \in [h]$);
(iv) output $mpk = (\{\boldsymbol{A}_i\}, \boldsymbol{u}, \{\boldsymbol{A}_{i,b,j}\}, \{\boldsymbol{B}_i\})$ and $msk = \{\boldsymbol{T}_{\boldsymbol{A}_i}\}$.

$\mathsf{Derive}(\boldsymbol{mpk}, \boldsymbol{sk_{t-1}}, \boldsymbol{v_1}, \cdots, \boldsymbol{v_t})$. On input of public-key mpk, secret key for the hierarchical level $t-1$ and vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_t$, the algorithm generates a secret key sk_t as follows:

(i) sample a new short basis for each lattice $\Lambda(\boldsymbol{F}_i || \boldsymbol{A}_{i,v_{t,i},t} + \boldsymbol{B}_i)_q^{\boldsymbol{u}}$, for $\boldsymbol{F}_i = [\boldsymbol{A}_i || \boldsymbol{A}_{i,v_{1,i},1} + \boldsymbol{B}_i || ... || \boldsymbol{A}_{i,v_{t-1,i},t-1} + \boldsymbol{B}_i]$ by involving $\boldsymbol{S}_i \leftarrow \mathsf{SampleBasisLeft}(\boldsymbol{F}_i, \boldsymbol{A}_{i,v_{t,i},t} + \boldsymbol{B}_i, \boldsymbol{S}'_i, \sigma)$, where $\boldsymbol{S}'_i \in sk_{t-1}$;
(ii) output $sk_t = \{\boldsymbol{S}_i\}$.

$\mathsf{Enc}(\boldsymbol{mpk}, \mathfrak{m}, \boldsymbol{w_1}, \cdots, \boldsymbol{w_t})$. On input of master public-key mpk, vectors $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_t$, and message $\mathfrak{m} \in \{0, 1\}$, the algorithm generates a ciphertext C as follows:

(i) choose a uniformly random vector $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n$, random matrices $\boldsymbol{R}_{i,j} \in \{-1, 1\}^{m \times m}$ and let $\beta = (l!)^2$;

(ii) choose a noise vector $\boldsymbol{x}_i \leftarrow \overline{\Psi}_{\alpha_t}^m$ and a noise term $x \leftarrow \overline{\Psi}_{\alpha_t}$;
(iii) calculate $\boldsymbol{F}_i = [\boldsymbol{A}_i || \boldsymbol{A}_{i,v_{1,i},1} + \boldsymbol{B}_i || ... || \boldsymbol{A}_{i,v_{t,i},t} + \boldsymbol{B}_i]$;
(iv) create l shares of vector $\boldsymbol{s}$ such that $\boldsymbol{s}_i = [p_1(i), ..., p_n(i)]$, for n random polynomials $p_i(x)$ of degree $r-1$, with $p_i(0) = s_i$
(v) compute $\boldsymbol{c}_i = \boldsymbol{F}_i^\top \boldsymbol{s}_i + \beta[\boldsymbol{x}_i || \boldsymbol{R}_i^\top \boldsymbol{x}_i] \in \mathbb{Z}_q^{(t+1)m}$, where $\boldsymbol{R}_i = [\boldsymbol{R}_{i,1} || ... || \boldsymbol{R}_{i,t}]$ and $c' = \boldsymbol{u}^\top \boldsymbol{s} + \beta \cdot x + \mathfrak{m} \cdot \lfloor q/2 \rceil \in \mathbb{Z}_q$;
(vi) output $C = (\{\boldsymbol{c}_i\}, c')$.

Dec($\boldsymbol{mpk}$, $\boldsymbol{sk_t}$, $\boldsymbol{C}$). On input of master public key mpk, secret key sk_t, and ciphertext C, the algorithm does the following:

(i) let $\mathbb{J}$ be the set of matching bits where $\boldsymbol{v}_{\gamma,i} = \boldsymbol{w}_{\gamma,i}$ for all $\gamma \in [t]$ and calculate each polynomial $l_j(x) = \prod_{i \in \mathbb{J}} \frac{x-i}{j-i}$;
(ii) the fractional Lagrangian coefficients will be $l_j = l_j(0)$ for $j \in \mathbb{J}$ so that $\sum l_j \boldsymbol{u}^\top \boldsymbol{s}_j = \boldsymbol{u}^\top \boldsymbol{s}$;
(iii) calculate each $\boldsymbol{e}_j$ by calling $\mathsf{SamplePre}(\boldsymbol{F}_j, \boldsymbol{S}_j, l_j \boldsymbol{u}, \sigma)$, where $\boldsymbol{S}_j \in sk_t$ and $\sigma = \sigma_t \sqrt{m(t+1)} \omega(\sqrt{\log(tm)})$;
(iv) compute $z = c' - \sum \boldsymbol{e}_j^\top \boldsymbol{c}_j \pmod q$; for $z \in (-q/2, q/2]$, output 0 if $|z| < q/4$ and 1 otherwise. Note that the threshold now is done between all vectors hierarchical, i.e., r must be the matching lines between matrices $\boldsymbol{V} = [\boldsymbol{v}_1^\top | ... | \boldsymbol{v}_t^\top]$ and $\boldsymbol{W} = [\boldsymbol{w}_1^\top | ... | \boldsymbol{w}_t^\top]$.

3.2. Correctness

If $v_{\gamma,i} = w_{\gamma,i}$, for all $v_{\gamma,i} \neq \star$, we have:

$$
\begin{aligned}
z &= c' - \sum \boldsymbol{e}_j^\top \boldsymbol{c}_j \pmod q \\
&= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \sum \boldsymbol{e}_j^\top (\boldsymbol{F}_j^\top \boldsymbol{s}_j + \beta[\boldsymbol{x}_j || \boldsymbol{R}_j^\top \boldsymbol{x}_j]) \pmod q \\
&= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \sum (\boldsymbol{F}_j \boldsymbol{e}_j)^\top \boldsymbol{s}_j - \sum \boldsymbol{e}_j^\top \beta[\boldsymbol{x}_j || \boldsymbol{R}_j^\top \boldsymbol{x}_j] \pmod q \\
&= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \sum l_j \boldsymbol{u}^\top \boldsymbol{s}_j - \sum \boldsymbol{e}_j^\top \beta[\boldsymbol{x}_j || \boldsymbol{R}_j^\top \boldsymbol{x}_j] \pmod q \\
&= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \boldsymbol{u}^\top \boldsymbol{s} - \sum \boldsymbol{e}_j^\top \beta[\boldsymbol{x}_j || \boldsymbol{R}_j^\top \boldsymbol{x}_j] \pmod q \\
&= \mathfrak{m} \cdot \lfloor q/2 \rceil + \underbrace{\beta \cdot x - \sum \boldsymbol{e}_j^\top \beta[\boldsymbol{x}_j || \boldsymbol{R}_j^\top \boldsymbol{x}_j]}_{\text{error term}} \pmod q
\end{aligned}
$$

3.3. Security Reduction

In this section we prove the following theorem.

Theorem 9. *If the decision-*$\mathsf{LWE}_{q,n,\chi}$ *problem is infeasible, then the HVE scheme described on Section 3.1 is IND-sAT-CPA .*

Proof.

Init. $\mathcal{B}$ is given lm LWE challenge pairs $(\boldsymbol{y}_{i,j}, z_{i.j}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ and a pair $(\boldsymbol{y}, z) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where, either $[z_{i,1}|...|z_{i,m}] = [\boldsymbol{y}_{i,1}|...|\boldsymbol{y}_{i,m}]^\top \boldsymbol{s} + \beta \boldsymbol{x}_i$ and $z = \langle \boldsymbol{y}, \boldsymbol{s} \rangle + \beta x$ for a random $\boldsymbol{s} \in \mathbb{Z}_q^n$, and noise terms $x \leftarrow \overline{\Psi}_{\alpha_t}$ and $\boldsymbol{x}_i \leftarrow \overline{\Psi}_{\alpha_t}^m$, or $z_{i,j}$ and z are uniformly random.

Setup. The public key is constructed using the vectors of the challenge pairs, as follows: $\boldsymbol{A}_i$ will be $\beta\boldsymbol{Y}$, where $\boldsymbol{Y} = [\boldsymbol{y}_{i,1}|...|\boldsymbol{y}_{i,m}]$, $\boldsymbol{A}_{i,b,j}$ will be $\boldsymbol{A}_i\boldsymbol{R}_{i,j} - \boldsymbol{B}_i$, each $\boldsymbol{R}_{i,j}$ is random in $\{-1, 1\}^m$, matrices $\boldsymbol{B}_i$ will be computed using $\mathsf{TrapGen}$ and $\boldsymbol{u}$ will be $\boldsymbol{y}$.

Secret keys. All private-key extraction queries are answered by using the trapdoor $\boldsymbol{T}_{\boldsymbol{B}_i}$ and the $\mathsf{SampleBasisRight}$ algorithm. It will output $sk_t = \{\boldsymbol{S}_i\}$, where $\boldsymbol{S}_i$ is a short basis for $\Lambda_q^{\perp}(\boldsymbol{A}_i\|\boldsymbol{A}_i\boldsymbol{R}_i + \boldsymbol{B}_i)$ by invoking

$$\boldsymbol{S}_i \leftarrow \mathsf{SampleBasisRight}(\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{R}_i, \boldsymbol{T}_{\boldsymbol{B}_i}, \boldsymbol{u}, \sigma).$$

Note that the distribution of the public parameters and keys in the real scheme is statistically indistinguishable from that in the simulation, as in [Agrawal et al. 2010] and [Abdalla et al. 2012].

Challenge Ciphertext. The algorithm now chooses a $\boldsymbol{s}'^{\star} \in \mathbb{Z}_q^n$ at random and calculates $\boldsymbol{s}^{\star}$ such that all shares $\boldsymbol{s}_i^{\star} = \boldsymbol{s}'^{\star}$, i.e., it solves the equation $\boldsymbol{s}_i^{\star} = \boldsymbol{s} + \sum \boldsymbol{a}_j i^j$ for $\boldsymbol{s}_i^{\star} = \boldsymbol{s}'^{\star}$, $i \in [1, l]$ and $j \in [0, r-1]$.

The construction of ciphertext $C = (\{\boldsymbol{c}_i\}, c')$ is based on the terms of the LWE challenges pairs, $c' = \beta w + \mathfrak{m}\lfloor q/2 \rceil$ and $\boldsymbol{c}_i$ is a concatenation of vectors $\beta\boldsymbol{R}_{i,j}^{\top}[z_{i,0}|...|z_{i,m}]$. If $[z_{i,0}|...|z_{i,m}] = \boldsymbol{Y}^{\top}\boldsymbol{s}'^{\star} + \boldsymbol{x}_i$ and $w = \langle \boldsymbol{u}, \boldsymbol{s}\rangle + x$ on the LWE challenges, then we have that the ciphertext is genuine:

$$\boldsymbol{c}_i = \boldsymbol{F}_i^{\top}\boldsymbol{s}'^{\star} + \beta[\boldsymbol{x}_i||\boldsymbol{R}_i^{\top}\boldsymbol{x}_i]$$

$$\boldsymbol{c}_i = [\boldsymbol{A}_i||\boldsymbol{A}_{i,v_{1,i},1} + \boldsymbol{B}_i||...||\boldsymbol{A}_{i,v_{t,i},t} + \boldsymbol{B}_i]^{\top}\boldsymbol{s}'^{\star} + \beta[\boldsymbol{x}_i||\boldsymbol{R}_i^{\top}\boldsymbol{x}_i]$$

$$\boldsymbol{c}_i = [\boldsymbol{A}_i||\boldsymbol{A}_i\boldsymbol{R}_{i,1} - \boldsymbol{B}_i + \boldsymbol{B}_i||...||\boldsymbol{A}_i\boldsymbol{R}_{i,t} - \boldsymbol{B}_i + \boldsymbol{B}_i]^{\top}\boldsymbol{s}'^{\star} + \beta[\boldsymbol{x}_i||\boldsymbol{R}_i^{\top}\boldsymbol{x}_i]$$

$$\boldsymbol{c}_i = [\boldsymbol{A}_i||\boldsymbol{A}_i\boldsymbol{R}_{i,1}||...||\boldsymbol{A}_i\boldsymbol{R}_{i,t}]^{\top}\boldsymbol{s}'^{\star} + \beta[\boldsymbol{x}_i||\boldsymbol{R}_i^{\top}\boldsymbol{x}_i]$$

$$\boldsymbol{c}_i = [\boldsymbol{A}_i^{\top}\boldsymbol{s}'^{\star} + \beta\boldsymbol{x}_i||\boldsymbol{R}_{i,1}^{\top}(\boldsymbol{A}_i^{\top}\boldsymbol{s}'^{\star} + \beta\boldsymbol{x}_i)||...||\boldsymbol{R}_{i,t}^{\top}(\boldsymbol{A}_i^{\top}\boldsymbol{s}'^{\star} + \beta\boldsymbol{x}_i)]$$

$$\boldsymbol{c}_i = [\beta(\boldsymbol{Y}^{\top}\boldsymbol{s}'^{\star} + \boldsymbol{x}_i)||\beta\boldsymbol{R}_{i,1}^{\top}(\boldsymbol{Y}^{\top}\boldsymbol{s}'^{\star} + \boldsymbol{x}_i)||...||\beta\boldsymbol{R}_{i,t}^{\top}(\boldsymbol{Y}^{\top}\boldsymbol{s}'^{\star} + \boldsymbol{x}_i)]$$

If $z_{i,j}$ and z are uniformly random, then the ciphertext is randomly generated.

Guess. $\mathcal{A}$ must guess whether it is interacting with a genuine or with a randomly generated ciphertext. The answer to this guess is also the answer to one of the LWE challenges. We showed that if z and all $z_{i,j}$ are uniformly random, then the ciphertext is randomly generated and if $[z_{i,1}|...|z_{i,m}] = [\boldsymbol{y}_{i,1}|...|\boldsymbol{y}_{i,m}]^{\top}\boldsymbol{s} + \beta\boldsymbol{x}_i$ and $w = \langle \boldsymbol{y}, \boldsymbol{s}\rangle + \beta x$, then the ciphertext is genuine. Therefore, $\mathcal{B}$'s advantage in solving LWE is the same as $\mathcal{A}$'s advantage in distinguishing whether the ciphertext is genuine or not.

□

4. Conclusion

In this paper we expand the basic fuzzy IBE scheme proposed by [Agrawal et al. 2012] into a hierarchical HVE scheme, in which users can generate secret keys, thus relieving the task of the Trusted Third Part. Our scheme is as secure as the original one based on the Learning With Errors Problem, while having the new feature. We also show a version using ideal lattices, that has the advantages of smaller key sizes and more efficient matrix multiplication, and how it affects the security proof.

References

Abdalla, M., De Caro, A., and Mochetti, K. (2012). Lattice-based hierarchical inner product encryption. In *LATINCRYPT 2012*, volume 7533 of *LNCS*, pages 121–138, Santiago, Chile. springer.

Agrawal, S., Boneh, D., and Boyen, X. (2010). Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572, French Riviera. springer.

Agrawal, S., Boyen, X., Vaikuntanathan, V., Voulgaris, P., and Wee, H. (2012). Functional encryption for threshold functions (or fuzzy ibe) from lattices. In *PKC 2012*, volume 7293 of *LNCS*, pages 280–297, Darmstadt, Germany. springer.

Agrawal, S., Freeman, D. M., and Vaikuntanathan, V. (2011). Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40, Seoul, South Korea. springer.

Alwen, J. and Peikert, C. (2009). Generating shorter bases for hard random lattices. In *STACS 2009*, pages 75–86.

Boneh, D., Sahai, A., and Waters, B. (2011). Functional encryption: Definitions and challenges. In *TCC 2011*, volume 6597 of *LNCS*, pages 253–273, Providence, RI, USA. springer.

Boneh, D. and Waters, B. (2007). Conjunctive, subset, and range queries on encrypted data. In *TCC 2007*, volume 4392 of *LNCS*, pages 535–554, Amsterdam, The Netherlands. springer.

Caro, A. D., Iovino, V., and Persiano, G. (2011). Hidden vector encryption fully secure against unrestricted queries. *IACR Cryptology ePrint Archive*, 2011:546.

Cash, D., Hofheinz, D., Kiltz, E., and Peikert, C. (2010). Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552, French Riviera. springer.

Gentry, C., Peikert, C., and Vaikuntanathan, V. (2008). Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206, Victoria, British Columbia, Canada. ACM Press.

Hanaoka, G., Nishioka, T., Zheng, Y., and Imai, H. (2009). An efficient hierarchical identity-based key-sharing method resistant against collusion-attacks. In *ASIACRYPT 2009*, volume 5479 of *LNCS*, pages 348–362, Cologne, Germany. springer.

Iovino, V. and Persiano, G. (2008). Hidden-vector encryption with groups of prime order. In *PAIRING 2008*, volume 5209 of *LNCS*, pages 75–88, Egham, UK. springer.

Katz, J., Sahai, A., and Waters, B. (2008). Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162, Istanbul, Turkey. springer.

Lyubashevsky, V., Peikert, C., and Regev, O. (2010). On ideal lattices and learning with errors over rings. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23, French Riviera. springer.

Micciancio, D. (2002). Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *FOCS 2002*, pages 356–365, Vancouver, British Columbia, Canada. IEEE.

Micciancio, D. and Regev, O. (2004). Worst-case to average-case reductions based on Gaussian measures. In *FOCS 2004*, pages 372–381, Rome, Italy. IEEE.

Pan, V. Y. (2001). *Structured matrices and polynomials: unified superfast algorithms*. Springer-Verlag New York, Inc., New York, NY, USA.

Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93, Baltimore, Maryland, USA. ACM Press.

Shamir, A. (1979). How to share a secret. *Commun. ACM*, 22(11):612–613.

Stehlé, D., Steinfeld, R., Tanaka, K., and Xagawa, K. (2009). Efficient public key encryption based on ideal lattices. In *ASIACRYPT 2009*, volume 5479 of *LNCS*, pages 617–635, Cologne, Germany. springer.

A. Ideal Lattice-Based HVE Scheme

In this section we provide the HVE scheme using ideal lattice, detailing the security proof and showing how it is affected by this change.

A.1. Our Construction

Let n be the security parameter, σ be the Gaussian parameter, l be the length of vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, r the threshold value and $\mathcal{R} = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$.

$\mathsf{Setup}(1^n)$. On input of security parameter n, the algorithm generates the public and secret keys as follows:

(i) run the $\mathsf{IdealTrapGen}(n, k, q, f, \sigma)$ algorithm to select uniformly $2l$ vectors $\hat{\boldsymbol{a}}_{i,b} \in \mathcal{R}^k$ (for $i \in [l]$ and $b \in \{0,1\}$), with a short basis $\boldsymbol{T}_{\hat{\boldsymbol{a}}_{i,b}} \in \mathbb{Z}^{kn \times kn}$ for $\Lambda_q^{\perp}(\boldsymbol{A}_{i,b})$, such that $\boldsymbol{A}_{i,b} = \mathsf{Rot}_f(\hat{\boldsymbol{a}}_{i,b})$;
(ii) choose uniformly random vector $\boldsymbol{u} \in \mathcal{R}$;
(iii) output $mpk = (\{\hat{\boldsymbol{a}}_{i,b}\}, \boldsymbol{u})$ and $msk = \{\boldsymbol{T}_{\hat{\boldsymbol{a}}_{i,b}}\}$.

$\mathsf{KeyGen}(\boldsymbol{mpk}, \boldsymbol{msk}, \boldsymbol{v})$. On input of public-key mpk, master key msk and vector $\boldsymbol{v}$, the algorithm generates a secret key sk as follows:

(i) choose n random polynomials $p_i(x)$ of degree $r-1$, with $p_i(0) = u_i$;
(ii) create l shares of vector $\boldsymbol{u}$ such that $\boldsymbol{u}_i = [p_1(i), ..., p_n(i)]$;
(iii) sample vectors for lattice $\Lambda(\boldsymbol{A}_{i,v_i})_q^{\boldsymbol{u}_i}$, with $\boldsymbol{A}_{i,v_i} = \mathsf{Rot}_f(\hat{\boldsymbol{a}}_{i,v_i})$ by invoking $\boldsymbol{e}_i \leftarrow \mathsf{SamplePre}(\boldsymbol{A}_{i,v_i}, \boldsymbol{T}_{\hat{\boldsymbol{a}}_{i,v_i}}, \boldsymbol{u}_i, \sigma) \in \mathbb{Z}^{kn}$ (for all i where $v_i = \star$, choose random $b \in \{0,1\}$ and use $\hat{\boldsymbol{a}}_{i,b}$);
(iv) output $sk = \{\boldsymbol{e}_i\}$.

$\mathsf{Enc}(\boldsymbol{mpk}, \mathfrak{m}, \boldsymbol{w})$. On input of master public-key mpk, vector $\boldsymbol{w}$, and message $\mathfrak{m} \in \{0,1\}$, the algorithm generates a ciphertext C as follows:

(i) choose a uniformly random vector $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and let $\beta = (l!)^2$;
(ii) choose a noise vector $\hat{\boldsymbol{x}}_i \leftarrow \overline{\Psi}_{\alpha_t}^{kn}$ and a noise term $x \leftarrow \overline{\Psi}_{\alpha_t}$;

(iii) compute $\boldsymbol{c}_i = \mathsf{Rot}_f(\hat{\boldsymbol{a}}_{i,w_i})^\top \boldsymbol{s} + \beta\hat{\boldsymbol{x}}_i \in \mathbb{Z}_q^{kn}$ and $c' = \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil \in \mathbb{Z}_q$;
(iv) output $C = (\{\boldsymbol{c}_i\}, c')$.

Dec($\boldsymbol{mpk}$, $\boldsymbol{sk}$, $\boldsymbol{C}$). On input of master public key mpk, secret key sk, and ciphertext C, the algorithm does the following:

(i) let $\mathbb{J}$ be the set of matching bits between vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, if $|\mathbb{J}| \geq r$; calculate each polynomial $l_j(x) = \prod_{i \in \mathbb{J}} \frac{x-i}{j-i}$;
(ii) the fractional Lagrangian coefficients will be $l_j = l_j(0)$ for $j \in \mathbb{J}$ so that $\sum l_j \boldsymbol{A}_{j,w_j} \boldsymbol{e}_j = \boldsymbol{u} \pmod q$;
(iii) compute $z = c' - \sum l_j \boldsymbol{e}_j^\top \boldsymbol{c}_j \pmod q$; for $z \in (-q/2, q/2]$, output 0 if $|z| < q/4$ and 1 otherwise.

A.2. Correctness

If $v_i = w_i$, for all $v_i \neq \star$, we have:

$$
\begin{aligned}
z &= c' - \sum l_j \boldsymbol{e}_j^\top \boldsymbol{c}_j \pmod q \\
z &= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \sum l_j \boldsymbol{e}_j^\top (\mathsf{Rot}_f(\hat{\boldsymbol{a}}_{j,w_j})^\top \boldsymbol{s} + \beta\hat{\boldsymbol{x}}_j) \pmod q \\
z &= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \sum l_j (\mathsf{Rot}_f(\hat{\boldsymbol{a}}_{j,w_j})\boldsymbol{e}_j)^\top \boldsymbol{s} - \sum l_j \boldsymbol{e}_j^\top \beta\hat{\boldsymbol{x}}_j \pmod q \\
z &= \boldsymbol{u}^\top \boldsymbol{s} + \beta x + \mathfrak{m} \cdot \lfloor q/2 \rceil - \boldsymbol{u}^\top \boldsymbol{s} - \sum l_j \boldsymbol{e}_j^\top \beta\hat{\boldsymbol{x}}_j \pmod q \\
z &= \mathfrak{m} \cdot \lfloor q/2 \rceil + \underbrace{\beta x - \sum l_j \boldsymbol{e}_j^\top \beta\hat{\boldsymbol{x}}_j}_{\text{error term}} \pmod q
\end{aligned}
$$

We have from Lemma 1 that $|\beta l_j| \leq \beta^2$, so we need to set the parameters in a way to guarantee that

$$\beta|x| + \sum \beta^2 |\boldsymbol{e}_j^\top \hat{\boldsymbol{x}}_j| < q/4$$

A.3. Security Reduction

In this section we prove the following theorem.

Theorem 10. *If the decision-*$\mathsf{ideal\text{-}LWE}_{q,n,\chi}$ *and decision-*$\mathsf{LWE}_{q,n,\chi}$ *problems are infeasible, then the HVE scheme described on Section A.1 is IND-sAT-CPA .*

Proof.

Init. $\mathcal{B}$ is given ideal-LWE l challenge pairs $(\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{z}}_i) \in \mathcal{R}^k \times \mathcal{R}^k$ and a LWE pair $(\boldsymbol{v}, w) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where, either $\hat{\boldsymbol{z}}_i = \mathsf{Rot}_f(\hat{\boldsymbol{y}}_i)^\top \boldsymbol{s} + \beta\hat{\boldsymbol{x}}_i$ and $z = \langle \boldsymbol{y}, \boldsymbol{s} \rangle + \beta x$ for a random $\boldsymbol{s} \in \mathbb{Z}_q^n$, and noise terms $x \leftarrow \overline{\Psi}_{\alpha_t}$ and $\hat{\boldsymbol{x}}_i \leftarrow \overline{\Psi}_{\alpha_t}^{kn}$, or $\hat{\boldsymbol{z}}_i$ and z are uniformly random.

Setup. The public key is constructed using the vectors of the challenge pairs, as follows: $\hat{\boldsymbol{a}}_{i,w_i}$ will be $\beta\hat{\boldsymbol{y}}_i$, $\hat{\boldsymbol{a}}_{i,\overline{w_i}}$ will be computed using IdealTrapGen and $\boldsymbol{u}$ will be $\boldsymbol{y}$.

Secret keys. All private-key extraction queries are answered using the following algorithm: (i) let $\mathbb{J}$ be the set of matching bits between vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ and $|\mathbb{J}| < r$; (ii) for all $j \in \mathbb{J}$, use algorithm $\mathsf{SampleGaussian}(\mathsf{Rot}_f(\hat{\boldsymbol{a}}_{j,w_j}), \sigma)$ to find vector $\boldsymbol{e}_j$, such that $\boldsymbol{u}_j = \mathsf{Rot}_f(\hat{\boldsymbol{a}}_{j,w_j})\boldsymbol{e}_j$; (iii) choose randomly $r - 1 - |\mathbb{J}|$ vectors $\boldsymbol{u}_i$; (iii) represent each

vector $\boldsymbol{u}_i$ as $\boldsymbol{u}_i = \boldsymbol{u} + \boldsymbol{a}_1 i + ... + \boldsymbol{a}_{r-1} i^{r-1}$, since we already calculated r vectors, it is possible to compute all vectors $\boldsymbol{u}_i$, for all $i \in [l]$. (iv) find the remaining vector $\boldsymbol{e}_i$ by calling the $\mathsf{SamplePre}(\mathsf{Rot}_f(\hat{\boldsymbol{a}}_{i,w_i}), \boldsymbol{T}_{\hat{\boldsymbol{a}}_{i,w_i}}, \boldsymbol{u}_i, \sigma)$ algorithm.

Note that the distribution of the public parameters and keys in the real scheme is statistically indistinguishable from that in the simulation.

Challenge Ciphertext. The ciphertext $C = (\{\boldsymbol{c}_i\}, c')$ is constructed based on the terms on the ideal-LWE challenges pairs, $c' = \beta z + \mathfrak{m}\lfloor q/2 \rceil$ and $\boldsymbol{c}_i = \beta \hat{\boldsymbol{z}}_i$. If $\hat{\boldsymbol{z}}_i = \mathsf{Rot}_f(\hat{\boldsymbol{y}}_i)^\top \boldsymbol{s} + \hat{\boldsymbol{x}}_i$ on the ideal-LWE challenge and $z = \langle \boldsymbol{u}, \boldsymbol{s} \rangle + x$ on the LWE challenge, then the ciphertext is genuine; if $\hat{\boldsymbol{z}}_i$ and z are uniformly random, then the ciphertext is randomly generated.

Guess. $\mathcal{A}$ must guess whether it is interacting with a genuine or with a randomly generated ciphertext. The answer to this guess is also the answer to one of the LWE challenges. We showed that if z and all $\hat{\boldsymbol{z}}_i$ are uniformly random, then the ciphertext is randomly generated and if $\hat{\boldsymbol{z}}_i = \mathsf{Rot}_f(\hat{\boldsymbol{y}}_i)^\top \boldsymbol{s} + \beta \hat{\boldsymbol{x}}_i$ and $z = \langle \boldsymbol{y}, \boldsymbol{s} \rangle + \beta x$, then the ciphertext is genuine. Therefore, $\mathcal{B}$'s advantage in solving ideal-LWE is the same as $\mathcal{A}$'s advantage in distinguishing whether the ciphertext is genuine or not.

□

A comparison of simple side-channel analysis countermeasures for variable-base elliptic curve scalar multiplication

Erick Nascimento[1], **Rodrigo Abarzúa**[2], **Julio López**[1], **Ricardo Dahab**[1]

[1] Institute of Computing, University of Campinas,
Av. Albert Einstein 1251, Campinas, Brazil.

ra032483@students.ic.unicamp.br, {jlopez, rdahab}@ic.unicamp.br
[2]Departamento de Matemática y Ciencia de la Computación,
Universidad de Santiago de Chile, Av. B. O'Higgins 3363, Santiago, Chile.

rodrigo.abarzua@usach.cl

Abstract. *Side-channel attacks are a growing threat to implementations of cryptographic systems. This article examines the state of the art of algorithmic countermeasures against simple side-channel attacks on elliptic curve cryptosystems defined over prime fields. We evaluate the security versus computation cost trade-offs of SSCA countermeasures for variable-base scalar multiplication algorithms without precomputation. The expected performance impact of each countermeasure is analyzed regarding their computational cost in terms of finite field operations.*

1. Introduction

Elliptic Curve Cryptography (ECC) is a class of public-key cryptosystems proposed by Neal Koblitz [Koblitz 1987] and Victor Miller [Miller 1985], which provides significant advantages in several situations, including implementations on specialized microprocessors. For example, some industry standards require 2048-bit integers [1] for the RSA system, whereas the equivalent security for ECC requires finite fields of just 160 bits. Given the restricted power consumption, storage and processing capacities of embedded microprocessors, ECC-based cryptosystems are an interesting option.

Passive side-channel attacks exploit physical leakages of a cryptographic process executing on a device, for example: timing [Kocher 1996], power consumption [Kocher et al. 1999] and electromagnetic radiation [Quisquater and Samyde 2001, Gandolfi et al. 2001]. These attacks present a realistic threat to cryptographic applications, and have been demonstrated to be very effective against smart cards without proper countermeasures [Mangard et al. 2007]. There are two general strategies for these attacks: *Simple Side-Channel Analysis* (SSCA) [Kocher 1996], which analyzes the measurements obtained during a single scalar multiplication, based on the differences in the measured quantity depending on the value of the secret key; and *Differential Side-channel Analysis* (DSCA) [Kocher et al. 1999], which is based on statistical techniques to retrieve information about the secret key based on measurements from several scalar multiplications.

[1]For the modulus n.

The aim of this paper is to show the landscape of solutions that implementers can choose to protect implementations of elliptic curve scalar multiplication algorithms against simple side-channel attacks, targeted at very restricted embedded devices. The SSCA countermeasures are evaluated from the security and computational cost (number of finite field operations) perspectives, providing a security versus computational cost comparison.

These tight device capabilities limited the scope of the paper to countermeasures to SSCA that require the minimum: additional data space at runtime, additional code space, time overhead and energy usage. Therefore, we have chosen countermeasures that do not make use of precomputation tables, usually to store elliptic curve points.

This paper is organized as follows. Section 2 provides an introduction to two kinds of simple side-channel analysis (SSCA): simple power analysis (SPA) and timing analysis (TA). Section 3 introduces the scalar multiplication problem and classic algorithms to solve it. Section 4 presents known variable-base scalar multiplication algorithms without precomputation and protected against SSCA, discussing their computational cost and known attacks. The performance comparison of the countermeasures is provided in Section 5. Finally, Section 6 concludes the paper.

2. Simple Side-channel Analysis (SSCA)

There are several types of side-channel analysis within the class of simple side-channel analysis, but two of them are commonly considered for software-based implementation of public-key cryptographic algorithms: simple power analysis (SPA) and timing analysis (TA).

2.1. Simple Power Analysis (SPA)

Power analysis in general, and simple power analysis in particular, exploit the fact that the instantaneous power consumption of a device depends on both: the data processed and the operation performed [Mangard et al. 2007, Kocher et al. 1999].

Power analysis countermeasures for both SPA and DPA are based on the reduction or elimination of the dependency between the power consumption of a cryptographic device and the intermediate values used by the algorithm, and are classified in two main groups: hiding and masking [Mangard et al. 2007].

The fundamental principle of *hiding* countermeasures is to remove the dependency of the data into the power consumption. In software implementations the goals are usually to randomize the algorithm control flow (time dimension) or the kind of instructions used on each run (amplitude dimension), without changing the input data or any other intermediate value processed by it, so that it is impossible to recognize the dependency between data and power consumption.

The concept of *masking* is to randomize the intermediate values processed by the cryptographic device, i.e., a masking operation is applied over these values before the original algorithm execution occurs. Sometime later, the resultant (masked) intermediate values are then unmasked. The goal of masking is to make the power consumption required to process the intermediate values on the masked implementation *independent* of the original (unmasked) values. To achieve this goal, masking countermeasures act on the amplitude dimension of the leakage, by means of data randomization.

2.2. Timing Analysis (TA)

Timing attacks against implementations of cryptographic algorithms exploit the fact that usually, in implementations, the elapsed time for the execution of an algorithm is variable and depends on the input data being processed on the particular run, be it fixed data (the key) or variable data (the plaintext).

In general, if an implementation is vulnerable to timing attacks it is also vulnerable to power attacks, but the converse is not necessarily true [Schindler 2002]. Timing analysis can often be combined with power analysis, to conceive powerful attacks. According to [Aciiçmez and Koç 2009], timing analysis can be classified in the following major groups: cache analysis, branch prediction analysis, and shared functional unit analysis.

3. Scalar Multiplication Algorithms

There are several classes of algorithms for multiplying a point P by an integer scalar k. In this paper we focus on variants of the double-and-add method. The double-and-add method is similar to square-and-multiplication method for modular exponentiation. The binary representation of k is denoted by $k_{n-1}2^{n-1} + \cdots + k_0 2^0$, then the scalar multiplication $[k]P = (k_{n-1}2^{n-1} + \cdots + k_0 2^0)P$ is computed using Horner's rule, resulting in the double-and-add method, $[k]P = [k_0 + 2(k_1 + 2(\ldots(k_{n-2} + 2k_{n-1})\ldots))]P$, which requires $n = \lfloor \log_2(k) \rfloor + 1$ point doublings and, in average, $\frac{n}{2}$ point additions $\left(nD + \frac{n}{2}A\right)$ [2]. The double-and-add method is optimal [Cohen et al. 2010].

4. Countermeasures for Variable-Base Scalar Multiplication without Precomputation against Simple Side-Channel Attacks

Elliptic curve scalar multiplication is particularly vulnerable to simple side-channel analysis because the operations of doubling and addition of points are intrinsically different. Very efficient countermeasures are known but they are only applicable to specific models of elliptic curves (e.g. Edwards curves [Bernstein et al. 2008, Hisil et al. 2008]). Although it is possible to select an elliptic curve from a model where efficient countermeasures are known, in practice, it is very likely that curves established by a standard will be selected. For example, NIST [NIST 2000] and SEC 2 [Certicom 2010] standards.

The most commonly used algorithm for computing $Q = [k]P$ on an elliptic curve is the *double-and-add* algorithm, in the *left-to-right* or *right-to-left* versions [3]. Suppose that the doubling of a point and the addition of two different points are implemented with different formulas. Then, these two operations can be distinguished by SSCA [Kocher 1996, Kocher et al. 1999]. When the power trace shows a point doubling followed by a point addition, the current bit, say k_i, is equal to 1; and when the power trace shows a doubling followed by another doubling, then $k_i = 0$. The usual approach to prevent SSCA consists in always repeating the same pattern of instructions, whatever the processed data is.

Several proposals have been made to protect scalar multiplication against these attacks. For example, the *double-and-add-always* algorithm of Coron [Coron 1999] ensures

[2] In this notation, D stands for point doubling and A stands for point addition.

[3] In the left-to-right version, the scalar bits are scanned from the most (MSB) to the least (LSB) significant bit. In the right-to-left version, the order is reversed.

the sequence of operations to compute a scalar multiplication is independent of the value of the secret scalar by inserting a dummy point addition between consecutive doublings (i.e., when the bit of the scalar is 0).

A second countermeasure is to use *unified formulas* which use similar sets of field operations for both additions and doubling operations. These formulas exist for Weierstrass curves [Brier and Joye 2002], and special curves, such as Edwards [Edwards 2007] and inverted Edwards [Bernstein and Lange 2007] curves, among others [4].

Another countermeasure is the *Montgomery ladder* [Montgomery 1987], a technique designed for a special type of curve in large characteristic fields. It makes sure that every bit of the scalar corresponds to both a doubling and an addition, and that both operations have an impact on the output of the scalar multiplication. The addition formula for curves on the Montgomery model is also much simpler than that for curves on the Weierstrass model, contributing to make the scalar multiplication faster in this curve model. However, it is not always possible to convert a curve in the Weierstrass model to one in the Montgomery model, because, among other reasons, the number of points in a Montgomery curve is always divisible by 4. Nevertheless, the converse is true [Cohen et al. 2010].

Elliptic curve cryptography standards recommend curves on the Weierstrass form over $\mathbb{F}_p$ (prime fields) or $\mathbb{F}_{2^m}$ (binary extension fields), where $p > 2$ is a prime number and m is an integer. None of the NIST recommended elliptic curves [NIST 2000] over prime fields can be converted to the Montgomery form, because all of them have a prime number of points (the cofactor is always 1).

A fourth approach consists in using *regular* representations of the scalar [Thériault 2006, Joye 2007], with the same fixed sequence of group operations for all scalars. Yet another countermeasure [Goundar et al. 2011] is the use of signed-digit representations of the scalar, particularly the zero-less signed-digit (ZSD) representation [5].

Finally, side-channel atomicity [Chevallier-Mames et al. 2004]) splits point operations into small homogeneous blocks of basic field operations, making it hard to distinguish between atomic blocks of point doublings from those of point additions.

The following countermeasures are considered in this paper: *a*) Unified Formulas of Brier-Joye and Brier-Dechene-Joye; *b*) Double-and-add-always of Coron; *c*) Montgomery Ladder over prime fields of Brier-Joye and Izu-Takagi; *d*) Double-add of Joye; *e*) Zero-less signed-digit (ZSD) of Goundar; and *f*) Atomic Blocks of Chevallier-Mames, Longa-Miri and Abarzúa-Thériault.

In the following subsections we present these countermeasures in detail, analyzing their side-channel security and the expected performance based on the number of required finite field operations.

[4] More details can be found in the database of special elliptic curves [Tanja and Bernstein 2014]. Such special families of elliptic curves are not studied in this work.

[5] An odd integer k is represented in ZSD if its digits are in the set $\{-1, 1\}$.

4.1. Unified Formulas of Brier-Joye [Brier and Joye 2002]

Unified Formulas for point addition and point doubling using projective coordinates were presented by Brier and Joye [Brier and Joye 2002]. Let $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, with $x_i = X_i/Z_i$ and $y_i = Y_i/Z_i$, then $R = P + Q = (X_3, Y_3, Z_3)$ is given by:

$$\begin{array}{lllll} U_1 = X_1Z_2, & U_2 = X_2Z_1, & S_1 = Y_1Z_2, & S_2 = Y_2Z_1, & T = U_1 + U_2, \\ M = S_1 + S_2, & Z = Z_1Z_2, & F = ZM, & R = T^2 - U_1U_2 + aZ^2, & L = MF, \\ G = TL, & W = R^2 - G, & X_3 = 2FW, & Y_3 = R(G - 2W) - L^2, & Z_3 = 2F^3. \end{array}$$

This formula requires $13M + 5S$ [6]. The Unified Formulas of Brier-Joye are prone to the following attacks.

4.1.1. Izu-Takagi Attack [Izu and Takagi 2002b]

The unified formulas of Brier-Joye are only valid if $y_1 + y_2 \neq 0$, where $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Izu and Takagi [Izu and Takagi 2002b] presented an attack using two points such that $x_1 \neq x_2$ and $y_1 + y_2 = 0$. The main idea of the attack is to use an exceptional point, which causes an exceptional condition ($0^{-1} \notin \mathbb{F}_p$, i.e., a division by zero) on the underlying unified formula in affine coordinates. The secret scalar k is guessed from the error of the scalar multiplication $[k]P$ for different points P. If an attacker wants to know if the target calculated $[m]P + P$ with $2 \leq m < k$, he can use a point P such that $y(mP) + y(P) = 0$. If the device replies an error to the attacker, or does it in an implicit but detectable manner, then he knows the device calculated $[m]P + P$. Starting with $m = 2$ [7], and by following this process, the attacker is able to recover the secret scalar bit-by-bit, from the most to the least significant.

Brier, Dechene and Joye [Brier et al. 2004] presented a new unified formula to protect against Izu-Takagi attack. Nevertheless, their formula is prone to both Amiel's attack [Amiel et al. 2009] and PACA attack [Amiel et al. 2007].

4.1.2. Walter's Attack [Walter 2004]

Walter's attack [Walter 2004] is based on the fact that the conditional subtraction in a Montgomery modular multiplication (MMM) operation can be detected. Given a point $P = (X, Y, Z)$, in the point doubling using the projective formulas of Brier-Joye the computation of U_1 and U_2 are identical ($U_1 = U_2 = XZ$), and they exhibit identical behavior with respect to the occurrence of the final conditional subtraction in MMM. The same property holds for the computation of S_1 and S_2.

The behavior for point addition is different. Point addition involves the input point $P = (X_1, Y_1, Z_1)$, where the random (or randomized) coordinates mean that occasionally

[6] In expressions regarding computational costs in terms of the number of finite field operations, M stands for (field) multiplication, S for squaring and I for inversion.

[7] When $m = 2$ and the attacker knows whether $y(2P) + y(P) = 0$, then, if it is, $k_{n-2} = 1$; otherwise, $k_{n-2} = 0$

X_1 and Y_1 will both be small (i.e., close to 0) and Z_1 will be large. This means that the computations of U_1 and S_1 are less likely to include the additional subtraction in MMM, while the computations of U_2 and S_2 are more likely to include the additional subtraction. This difference in behavior can be detected by an attacker and can be used accordingly to recover the bits of the secret scalar.

4.1.3. Amiel *et al*'s Attacks [Amiel et al. 2009]

A common requirement for several countermeasures against simple side-channel attacks is that multiplication and squaring field operations are indistinguishable from the side-channel analysis point of view, i.e., both squaring and multiplication must be computed using the same algorithm. Particularly, atomic blocks [Chevallier-Mames et al. 2004, Chen et al. 2009, Giraud and Verneuil 2010] and *Unified formulas* [Brier and Joye 2002, Bernstein and Lange 2007, Joye et al. 2010] countermeasures rely on this property. However, that's not usually the case. Amiel's attack [Amiel et al. 2009] is based on distinguishing between multiplications and squarings using the instantaneous power consumption trace. This is possible because the Hamming weight probability distribution of the result of a multiplication is distinct from that of a squaring operation, and they can be distinguished in the power traces.

Notice that the computation of multiplications $Z = Z_1 Z_2$ and $U_1 U_2$ in $R = T^2 - U_1 U_2 + aZ^2$, when a point addition is performed, are different from those when a point doubling operation is performed. The computation of $Z = Z_1^2$ and U_1^2 in R will be squaring operations and, hence, Amiel's attack can be applied to such implementations.

4.1.4. Passive and Active Combined Attack (PACA) [Amiel et al. 2007]

Amiel *et al.* [Amiel et al. 2007] presented a Passive and Active Combined Attack (PACA) on a (supposedly) side channel resistant implementation of the square and multiply algorithm. Although not a pure SSCA attack, because of the required fault insertion step (the active part of the attack), we present it here for completeness.

The main idea of the PACA attack is as follows. An attacker applies a fault in the register storing the Z coordinate of point P_1, say setting $Z_1 = 0$ after the fault. Then, in the Unified formulas of Brier-Joye, we have two different patterns for the calculation, $Z = Z_1 \cdot Z_1 = 0 \cdot 0$ (if it is a doubling) and $Z = Z_1 \cdot Z_2 = 0 \cdot Z_2$ (with $Z_2 \neq 0$, if it is an addition), and both can be identified in a power trace [Amiel et al. 2007, Schmidt et al. 2010]. This allows an SSCA attacker to distinguish between point doubling and addition operations and consequently to recover the secret scalar.

4.2. Double-and-add-always algorithm of Coron [Coron 1999]

The *double-and-add-always* algorithm of Coron [Coron 1999] (Algorithm 1) uses a dummy point addition when the scalar bit k_i is 0, such that the sequence of operations to compute the scalar multiplication is independent of the value of the secret scalar.

Therefore an adversary cannot guess the bit k_i by SPA. A drawback of this method is its low efficiency. It requires $nA + nD$ field operations, a 33% increase in the amount of field operations in comparison to the (unprotected) binary left-to-right algorithm.

Algorithm 1 *Double-and-add always* algorithm resistant against SPA

INPUTS: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
OUTPUTS: $Q = [k] \cdot P$
1: $R_0 \leftarrow P_\infty$
2: **for** i **from** $n-1$ **to** 0 **do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad R_1 \leftarrow R_0 + P$
5: $\quad R_0 \leftarrow R_{k_i}$
6: **end for**
7: return R_0

The *Double-and-add always* algorithm of Coron is prone to the following attacks.

4.2.1. Fouque and Valette's Doubling Attack [Fouque and Valette 2003]

The doubling attack of Fouque-Valette [Fouque and Valette 2003] is based on the fact that it is possible to detect if two intermediate values are equal when the algorithm computes the scalar multiplication for points chosen points P and $2P$. Several algorithms protected against SPA are vulnerable to Fouque and Valette's attack, such as the classic binary left-to-right algorithm, including those derived from it, such as Coron's double-and-add-always algorithm.

In Coron's double-and-add-always algorithm (Algorithm 1), the partial sums are computed as follows: $S_m(P) = \sum_{i=1}^{m} k_{n-i} 2^{m-i} P = \sum_{i=1}^{m-1} k_{n-i} 2^{m-1-i}(2P) + k_{n-m}P = S_{m-1}(2P) + k_{n-m}P$. So, the intermediate result of the algorithm at step m when given input P will be equal to the intermediate result at step $m-1$ when given input $2P$, if and only if, $k_{n-m} = 0$. Therefore, an attacker can obtain the secret scalar by comparing the doubling computation at step $m+1$ for P and at step m for $2P$ to recover the bit k_{n-m}. If both computations are identical, $k_{n-m} = 0$, otherwise $k_{n-m} = 1$. It has been shown that with only two scalar multiplication requests chosen by the attacker, it is possible to recover all the bits of the scalar [8].

4.3. Montgomery Ladder of Brier-Joye [Brier and Joye 2002]

Another possible countermeasure is the *Montgomery ladder* method [Montgomery 1987], originally designed for a special type of curve, the so-called Montgomery curve in large characteristic. Brier and Joye [Brier and Joye 2002] extended this method to Weierstrass curves of large characteristic. Their algorithm requires $9M + 2S$ for point addition and $6M+3S$ for point doubling. The classic Montgomery powering ladder is prone to M safe-error fault attacks [9] [Sung-Ming et al. 2002], Joye and Yen [Joye and Yen 2003] proposed modifications in order to counteract them (Algorithm 2).

The modified Montgomery ladder makes it impossible to insert safe faults, thus providing a natural protection against SPA, M safe-error and C safe-error attacks [10] [Joye and Yen 2003]. An important observation is that this algorithm al-

[8] The attacker collects one power trace for the computation of kP and one for the computation of $k(2P)$. For each iteration $m = 1, ..., n$, he runs the attack as described and finds k_{n-m}.

[9] A M safe-error is a memory safe-error in which bits of a register are maliciously modified, and the change is temporary, i.e. the register can be overwritten later.

[10] A C safe-error stands for computational safe-error, and consists in timely induce a temporary fault in the ALU for determining whether an operation is dummy or effective.

lows one to compute scalar multiplication on elliptic curves using the x-coordinate only [Brier and Joye 2002, Fischer and Giraud 2002, Izu and Takagi 2002a]. In Table 1 we present the cost of this countermeasure, including its refined forms.

Algorithm 2 Montgomery ladder resistant against SPA and safe fault attacks

INPUTS: A point $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$
OUTPUTS: $Q = [k] \cdot P$
1: $R_0 \leftarrow P_\infty$, $R_1 \leftarrow P$
2: **for** i **from** $n-1$ **to** 0 **do**
3: $\quad b \leftarrow k_i$,
4: $\quad R_{1-b} \leftarrow R_{1-b} + R_b$
5: $\quad R_b \leftarrow 2R_b$
6: **end for**
7: **return** R_0

Table 1. Computing cost for algorithms based on Montgomery ladder

Algorithm	In	# regs.	Total cost
Classic Montgomery ladder of Brier-Joye	[Brier and Joye 2002]	8	$n(12M + 13S) + 1I + 3M + 1S$
$X-$only Montgomery ladder	[Brier and Joye 2002, Izu and Takagi 2002a]	7	$n(9M + 7S) + 1I + 14M + 3S$
$(X, Y)-$only co$-Z$ Montgomery ladder	Alg. 15 in [Goundar et al. 2011]	6	$n(8M + 6S) + 1I + 1M$

The Montgomery Ladder of Brier-Joye is prone to the following attacks.

4.3.1. Relative Doubling Attack of Yen *et al* [Yen et al. 2006]

Yen *et al.* [Yen et al. 2006] proposed the relative doubling attack, which uses the same chosen input (P and $2P$) as described in Fouque and Valettes's doubling attack (Section 4.2.1). In this attack it is just required to determine the relation between two adjacent secret scalar bits (i.e., if $k_i = k_{i-1} = 0$ or $k_i = k_{i-1} = 1$ holds), thereby decreasing the number of key candidates.

4.4. Double-add algorithm of Joye [Joye 2007]

Joye's double-add algorithm [Joye 2007] (Algorithm 3), like Montgomery ladder for right-to-left scalar multiplications, always repeats the same pattern of effective operations. Table 2 shows the cost for the classic Joye's double-add algorithm and the variant using the Co-Z technique [Goundar et al. 2011]. There is no known SSCA attack against this algorithm.

Algorithm 3 Joye's double-add resistant against SPA

INPUTS: A point $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
OUTPUTS: $Q = [k] \cdot P$
1: $R_0 \leftarrow P_\infty$, $R_1 \leftarrow P$
2: **for** i **from** 0 **to** $n-1$ **do**
3: $\quad b \leftarrow k_i$,
4: $\quad R_{1-b} \leftarrow 2R_{1-b} + R_b$
5: **end for**
6: **return** R_0

Table 2. Computing cost of Joye's double-add

Algorithm	In	# regs.	Total cost
Classic Joye's double-add	Alg. 5 in [Joye 2007]	10	$n(13M + 8S) + 1I + 3M + 1S$
Co-Z Joye's double-add	Alg. 14 in [Goundar et al. 2011]	8	$n(9M + 7S) + 1I - 9M - 6S$

4.5. Signed Digit Methods of Goundar *et al.* [Goundar et al. 2011]

In order to prevent SPA-type attacks, Goundar *et al* [Goundar et al. 2011] proposed the use of the *zeroless signed-digit expansion* (ZSD) in the binary left-to-right or right-to-left algorithms. The odd scalar k, is recoded with digits in the set $\{-1, 1\}$. The recoding is on-the-fly, taking place inside the main loop (Algorithm 4).

The computational costs for the most efficient algorithms using the signed digit method are shown in Table 3: the right-to-left signed-digit algorithm using the Co-Z technique and the corresponding left-to-right algorithm. The latter also applies the (X,Y)-only technique, which simplifies some of the curve operations, rendering the Z coordinate (in projective coordinates) unnecessary and thus improving the cost.

Algorithm 4 Classic Signed-digit method: Left-to-right

INPUTS: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$ with $k_0 = 1$
OUTPUTS: $\mathbf{Q} = [k] \cdot \mathbf{P}$
1: $R_0 \leftarrow P$; $R_1 \leftarrow P$
2: **for** i **from** $n - 1$ **to** 1 **do**
3: $\quad \kappa \leftarrow (-1)^{1+k_i}$
4: $\quad R_0 \leftarrow 2R_0 + (\kappa)R_1$
5: **end for**
6: **return** R_0

Table 3. Computing cost of Signed-digit method

Algorithm	In	# regs.	Total cost
Right-to-left algorithm			
Co$-Z$ signed-digit algorithm	Alg. 17 in [Goundar et al. 2011]	8	$n(9M + 7S) + 1I - 9M - 6S$
Left-to-right algorithm			
$(X, Y)-$only co$-Z$ signed-digit algorithm	Alg. 16 in [Goundar et al. 2011]	6	$n(8M + 6S) + 1I - 5M - 4S$

4.6. Atomic Blocks of Chevallier-Mames *et al.* [Chevallier-Mames et al. 2004]

Atomic blocks [Chevallier-Mames et al. 2004] is a method to secure scalar multiplication against SSCA consisting in partitioning point operations into small homogeneous atomic blocks, which cannot be distinguished from each other through SSCA. The original atomic block of Chevallier-Mames has a (M, A, N, A) [11] structure of field operations.

One important assumption was made in the atomic blocks of Chevallier-Mames: multiplication and squaring are indistinguishable from a side-channel perspective. This was later proved wrong by [Amiel et al. 2009] (see Section 4.1.3) and [Hanley et al. 2011].

Longa and Miri [Longa and Miri 2008] presented a new atomic block structure based on the sequence (S, N, A, M, N, A, A) [12] of field operations. Their atomic block

[11] Multiplication-Addition-Negation-Addition.
[12] Squaring-Negation-Addition-Multiplication-Negation-Addition-Addition.

Table 4. Costs of scalar multiplication using the atomic blocks of Abarzúa and Thériault [Abarzúa and Thériault 2012]

Algorithm	In	# regs.	Total cost
Right-to-left algorithm			
Modified Doubling and General Addition	[Abarzúa and Thériault 2012]	16	$n(8.5M + 8.5S) + 1I + 3M + 1S$
Left-to-right algorithm			
Doubling and Mixed Addition	[Abarzúa and Thériault 2012]	11	$n(7M + 7S) + 1I + 3M + 1S$

structure have been applied to doubling, tripling and mixed addition for elliptic curves in Jacobian coordinates.

Abarzúa and Thériault [Abarzúa and Thériault 2012] built new sets of atomic blocks designed to protect against both SSCA and C-safe fault attacks. These atomic blocks are structured with the sequence of field operations (S, N, A, A, M, A). They applied these atomic blocks to various operations in Jacobian coordinates: doubling, tripling, and quintupling, as well as mixed Jacobian-affine addition. Formulas were also given to the general Jacobian addition and Modified-Jacobian doubling for use in right-to-left scalar multiplication. Finally, they presented a variation of the Jacobian doubling formula that requires the same number of blocks as the mixed Jacobian-affine addition, essentially giving the atomic equivalent of unified formulas.

Table 4 summarizes the scalar multiplication costs using these atomic blocks in the right-to-left and left-to-right algorithms.

5. Computational cost versus security comparison

For the computational cost comparison between different algorithms, we consider the following cost ratio for the finite field operations: $S/M = 0.8$ and $I/M = 100$. We also consider that the scalar k is $n = 192$ bits in length. Tables 5 and 6 summarizes the expected (theoretical) computational cost and the security issues of the different countermeasures for scalar multiplication algorithms against SSCA.

Among the right-to-left algorithms, Joye's double-add and Goundar's signed-digit algorithm are tied as the most efficient. Abarzúa and Thériault's atomic blocks is the most efficient left-to-right (and overall, in fact) scalar multiplication algorithm protected against SSCA, and there is not any known attack against it.

Table 5. Comparison of protected *left-to-right* scalar multiplication algorithms

Countermeasure	**Coordinate Systems**	**Total Cost**	**Performance** $n = 192$	**Security Problem**
Unified Formulas for Weierstrass curves	$\mathcal{P}$	$n(13M + 5S) + 1I + 2M$	$3366M$	(ψ)
		$n(16M + 3S) + 1I + 2M$	$3634.8M$	(ϕ)
Double-and-Add-Always	$\mathcal{J}$	$n(10M + 9S) + 1I + 3M + 1S^{(\mathbf{a})}$	$3406.2M$	(φ)
Montgomery Ladder for Weierstrass curves	$\mathcal{J}$	$n(8M + 6S) + 1I + 1M^{(\mathbf{b})}$	$2558.6M$	-
		$n(9M + 7S) + 1I + 14M + 3S^{(\mathbf{c})}$	$2919.6M$	-
Signed-digit	$\mathcal{J}$	$n(8M + 6S) + 1I - 5M - 4S^{(\mathbf{d})}$	$2549.4M$	-
Atomic Blocks	$\mathcal{J}$	$n(7M + 7S) + 1I + 3M + 1S^{(\mathbf{e})}$	$2523M$	(ξ)

Detailed description of computational cost:

(a) [Abarzúa and Thériault 2012]: fast mixed addition $(7M + 4S)$ and fast doubling $(3M + 5S)$ with $a = -3$.
(b) [Goundar et al. 2011]: (X, Y)-only co-Z Montgomery ladder, $(8M + 6S)$ for each bit.
(c) [Brier and Joye 2002, Izu and Takagi 2002a]: X-only Montgomery ladder, $(9M + 7S)$ for each bit.
(d) [Goundar et al. 2011]: $(X, Y)-$only co-Z signed-digit algorithm, $(8M + 6S)$ for each bit.
(e) [Abarzúa and Thériault 2012]: addition $(6M + 6S)$ and doubling $(4M + 4S)$ [13].

Attacks:

(ψ) [Izu and Takagi 2002b, Walter 2004, Amiel et al. 2009, Schmidt et al. 2010].
(ϕ) [Stebila and Thériault 2006, Amiel et al. 2009, Amiel et al. 2007].
(φ) [Fouque and Valette 2003]: doubling attack.
(ξ) [Fouque and Valette 2003, Chen et al. 2009] [14]: these attacks do not apply to Abarzúa-Thériault atomic blocks.

Table 6. Comparison of protected right-to-left scalar multiplication algorithms

Countermeasure	Coordinate Systems	Total Cost	Performance $n = 192$	Security Problem
Joye's double-add	$\mathcal{J}$	$n(9M + 7S) + 1I - 9M - 6S^{(\mathbf{f})}$	$2889.4M$	-
Signed-digit	$\mathcal{J}$	$n(9M + 7S) + 1I - 9M - 6S^{(\mathbf{g})}$	$2889.4M$	-
Atomic Blocks	$\mathcal{J}$	$n(8.5M + 8.5S) + 1I + 3M + 1S^{(\mathbf{h})}$	$3041.4M$	(χ)

Detailed description of computational cost:

(f) [Goundar et al. 2011]: Co-Z Joye's double-add, $(9M + 7S)$ for each bit.
(g) [Goundar et al. 2011]: Co-Z signed-digit algorithm, $(9M + 7S)$ for each bit.
(h) [Abarzúa and Thériault 2012]: general addition $(9M+9S)$ and doubling $(4M+4S)$.

Attacks:

(χ) Chen's attack [Chen et al. 2009] [15]. This attack does not apply to Abarzúa-Thériault atomic blocks.

6. Conclusion and future work

Side-channel attacks are a growing threat to implementations of cryptographic systems. This article examined the state of the art of algorithmic countermeasures for variable-base scalar multiplication algorithms without precomputation. A comparison has been made between several classes of proposed countermeasures regarding their computational costs, claimed security properties and known attacks.

[13] In this case the algorithm performs $nD + \frac{n}{2}A$.
[14] These attacks works because the implementation does not avoid irregular breaks between atomic blocks within the same group operation and distinct group operations.
[15] This experimental attack applies because the implementation does not avoid irregular breaks between atomic blocks within the same group operation and distinct group operations.

It is known that theoretical computational costs based on finite field operation counts does not tell which algorithm is in fact the most efficient in practice. Implementations on the target platform are required to have a true picture of their real performance. This is even more true when side-channel protected implementations are required, because implementation-level protections (e.g., against timing analysis) for one algorithm may be more costly than those for another algorithm, as they may be dependent on the structure of the algorithm.

Besides the performance results from a real implementation, a security assessment of the selected algorithms is also required.

References

Abarzúa, R. and Thériault, N. (2012). Complete Atomic Blocks for Elliptic Curves in Jacobian Coordinates over Prime Fields. In Hevia, A. and Neven, G., editors, *Progress in Cryptology – LATINCRYPT 2012*, volume 7533 of *LNCS*, pages 37–55. Springer Berlin / Heidelberg.

Aciiçmez, O. and Koç, c. K. (2009). Microarchitectural Attacks and Countermeasures. In *Cryptographic Engineering*, chapter 18. Springer.

Amiel, F., Feix, B., Tunstall, M., Whelan, C., and Marnane, W. P. (2009). Distinguishing multiplications from squaring operations. In *Selected Areas in Cryptography*, pages 346–360. Springer.

Amiel, F., Villegas, K., Feix, B., and Marcel, L. (2007). Passive and active combined attacks: Combining fault attacks and side channel analysis. In *Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on*, pages 92–102.

Bernstein, D. and Lange, T. (2007). Inverted edwards coordinates. In Boztaş, S. and Lu, H.-F., editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 4851 of *LNCS*, pages 20–27. Springer.

Bernstein, D. J., Birkner, P., Joye, M., Lange, T., and Peters, C. (2008). Twisted edwards curves. In *Progress in Cryptology–AFRICACRYPT 2008*, pages 389–405. Springer.

Brier, E., Dechene, I., and Joye, M. (2004). Unified Point Addition Formulae for Elliptic Curve Cryptosystems. In *Embedded Cryptographic Hardware: Methodologies and Archi- tectures*, pages 247–256. Nova Science Publishers.

Brier, E. and Joye, M. (2002). Weierstrass Elliptic Curves and Side-Channel Attacks. In *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems, vol 2274*, pages 335–345. Springer.

Certicom (2010). SEC 2: Recommended Elliptic Curve Domain Parameters, version 2.0. Technical report, Certicom Corp.

Chen, T., Li, H., Wu, K., and Yu, F. (2009). Countermeasure of ECC against Side-channel Attacks: Balanced Point Addition and Point Doubling Operation Procedure. *Asia-Pacific Conference on Information Processing*.

Chevallier-Mames, B., Ciet, M., and Joye, M. (2004). Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity. *Computers, IEEE Transactions on*, 53(6):760–768.

Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., and Vercauteren, F. (2010). *Handbook of elliptic and hyperelliptic curve cryptography*. Chapman and Hall/CRC.

Coron, J.-S. (1999). Resistance against differential power analysis for elliptic curve cryptosystems. In *Cryptographic Hardware and Embedded Systems*, pages 292–302. Springer.

Edwards, H. (2007). A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44(3):393–422.

Fischer, W. and Giraud, C. (2002). Parallel scalar multiplication on general elliptic curves over Fp hedged against Non-Differential Side-Channel Attacks. *IACR Cryptology ePrint Archive*.

Fouque, P.-A. and Valette, F. (2003). The doubling attack – why upwards is better than downwards. In Walter, C., Koç, e., and Paar, C., editors, *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *LNCS*, pages 269–280. Springer.

Gandolfi, K., Mourtel, C., and Olivier, F. (2001). Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES 2001*, pages 251–261. Springer.

Giraud, C. and Verneuil, V. (2010). Atomicity improvement for elliptic curve scalar multiplication. In Gollmann, D., Lanet, J.-L., and Iguchi-Cartigny, J., editors, *Smart Card Research and Advanced Application*, volume 6035 of *LNCS*, pages 80–101. Springer.

Goundar, R., Joye, M., Miyaji, A., Rivain, M., and Venelli, A. (2011). Scalar multiplication on Weierstraßelliptic curves from Co-Z arithmetic. *Journal of Cryptographic Engineering*, 1(2):161–176.

Hanley, N., Tunstall, M., and Marnane, W. (2011). Using templates to distinguish multiplications from squaring operations. *International Journal of Information Security*, 10(4):255–266.

Hisil, H., Wong, K. K.-H., Carter, G., and Dawson, E. (2008). Twisted Edwards curves revisited. In *Advances in Cryptology-ASIACRYPT 2008*, pages 326–343. Springer.

Izu, T. and Takagi, T. (2002a). A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks. In *Public Key Cryptography – PKC 2002*, pages 280–296.

Izu, T. and Takagi, T. (2002b). Exceptional procedure attack on elliptic curve cryptosystems. In *Public Key Cryptography—PKC 2003*, pages 224–239. Springer.

Joye, M. (2007). Highly regular right-to-left algorithms for scalar multiplication. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 135–147. Springer.

Joye, M., Tibouchi, M., and Vergnaud, D. (2010). Huff's model for elliptic curves. In *Algorithmic Number Theory*, pages 234–250. Springer.

Joye, M. and Yen, S.-M. (2003). The Montgomery powering ladder. In *Cryptographic Hardware and Embedded Systems-CHES 2002*, pages 291–302. Springer.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209.

Kocher, P. (1996). Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Koblitz, N., editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer Berlin / Heidelberg.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential Power Analysis. In Wiener, M., editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, page 789. Springer Berlin / Heidelberg.

Longa, P. and Miri, A. (2008). Fast and Flexible Elliptic Curve Point Arithmetic over Prime Fields. In *Computers, IEEE Transactions on*, volume 57, pages 289–302.

Mangard, S., Oswald, E., and Popp, T. (2007). *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer.

Miller, V. S. (1985). Use of Elliptic Curves in Cryptography. In *Advances in Cryptology -CRYPTO Proceedings*, pages 417–426. Springer.

Montgomery, P. L. (1987). Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264.

NIST (2000). FIPS 186-2: Digital Signature Standard. Technical report, NIST.

Quisquater, J.-J. and Samyde, D. (2001). Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security*, pages 200–210. Springer.

Schindler, W. (2002). A Combined Timing and Power Attack. In Naccache, D. and Paillier, P., editors, *Public Key Cryptography SE - 19*, volume 2274 of *LNCS*, pages 263–279. Springer.

Schmidt, J.-M., Tunstall, M., Avanzi, R., Kizhvatov, I., Kasper, T., and Oswald, D. (2010). Combined implementation attack resistant exponentiation. In *Progress in Cryptology–LATINCRYPT 2010*, pages 305–322. Springer.

Stebila, D. and Thériault, N. (2006). Unified point addition formulæ and side-channel attacks. In Goubin, L. and Matsui, M., editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 354–368. Springer.

Sung-Ming, Y., Kim, S., Lim, S., and Moon, S. (2002). A countermeasure against one physical cryptanalysis may benefit another attack. In Kim, K., editor, *Information Security and Cryptology — ICISC 2001*, volume 2288 of *LNCS*, pages 414–427. Springer.

Tanja, L. and Bernstein, D. J. (2014). Explicit-Formulas Database. www.hyperelliptic.org/EFD/bib.html.

Thériault, N. (2006). Spa resistant left-to-right integer recodings. In Preneel, B. and Tavares, S., editors, *Selected Areas in Cryptography*, volume 3897 of *LNCS*, pages 345–358. Springer.

Walter, C. (2004). Simple power analysis of unified code for ecc double and add. In Joye, M. and Quisquater, J.-J., editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 191–204. Springer.

Yen, S.-M., Ko, L.-C., Moon, S., and Ha, J. (2006). Relative doubling attack against montgomery ladder. In Won, D. and Kim, S., editors, *Information Security and Cryptology - ICISC 2005*, volume 3935 of *LNCS*, pages 117–128. Springer.

Sistema Indicador de Resiliência na Conectividade de Redes Heterogêneas sem fio

Robson Melo, Michele Nogueira, Aldri Santos

[1]Núcleo de Redes Sem Fio e Redes Avançadas (NR2)
Universidade Federal do Paraná – Curitiba – Brasil

{rgmelo,michele,aldri}@inf.ufpr.br

Abstract. *The dynamics and complexity of heterogeneous wireless networks difficult the development of adaptive security solutions to their conditions. The convergence of several communication technologies improves the connectivity, but they can also damage the availability of services on network. In this way, the network resilience is needed to tolerate the frequent disconnections, supporting the connectivity for several technologies. This paper presents an indicator system of security for heterogeneous wireless wireless networks sensitive fragility and robustness of connectivity. The system evaluates the criticality of communication links and redundant routes between devices on the network to indicate the degree of security of the different access networks in an environment of overlapping heterogeneous networks. The results show that evaluation of availability of connectivity were accurate and that the proposal can be applied in different types of networks, supporting the creation of solutions focused on resilience and security in heterogeneous wireless networks.*

Resumo. *A dinamicidade e a complexidade das redes heterogêneas sem fio dificultam o desenvolvimento de soluções de segurança adaptativas às suas condições. A convergência de diferentes tecnologias de comunicação amplia as opções de conectividade, contudo, agregada aos problemas de segurança, podem afetar a disponibilidade dos serviços na rede. Soluções que garantam resiliência à rede são necessárias para tolerar as frequentes desconexões dos terminais móveis e ainda assim garantir a disponibilidade de conectividade por diferentes tecnologias de comunicação. Este trabalho apresenta um sistema indicador de resiliência, particularmente fragilidade e robustez, na conectividade de redes heterogêneas sem fio. O sistema avalia a criticidade dos enlaces de comunicação e a redundância de rotas entre os dispositivos na rede para indicar o grau de resiliência de diferentes redes de acesso em um ambiente de redes heterogêneas sobrepostas. Os resultados mostram a efetividade do sistema em avaliar a fragilidade e a robustez da conectividade e a aplicação do mesmo em diferentes tipos de redes, suportando a criação de soluções voltadas à resiliência e a segurança em redes heterogêneas sem fio.*

1. Introdução

A popularização dos dispositivos computacionais portáteis e o aumento da disponibilidade das redes de acesso têm intensificado o desejo dos usuários por conectividade em qualquer lugar e a todo instante [Melo et al. 2013]. As redes heterogêneas, formadas por diferentes redes com tipo de tecnologias de comunicação distintas, buscam atender a essa

demanda. Contudo, a convergência dessas diferentes tecnologias de comunicação traz consigo alguns problemas, principalmente sob o aspecto de segurança. As vulnerabilidades de segurança restritas a apenas um tipo de rede pode intensificar as vulnerabilidades de outras tecnologias além de se proliferar mais facilmente [Ghosh et al. 2012].

As soluções de segurança concebidas de forma individualizada para cada tipo de rede nem sempre são eficazes e necessitam ser repensadas quando aplicadas em redes heterogêneas [He et al. 2013]. A convergência de diferentes tecnologias de comunicação amplia as opções de conectividade, mas os problemas de segurança podem comprometer diretamente a disponibilidade dos serviços na rede e principalmente sua conectividade [Melo et al. 2013]. Em redes heterogêneas sem fio, o serviço de *handoff*, que corresponde à transição de conexão de uma rede para outra, pode sofrer interferências e ações de usuários maliciosos e acabar por indisponibilizar a comunicação [He et al. 2013]. Os dispositivos móveis conectados em uma rede *mesh* sem fio que migram suas conexões para redes *celulares 3G*, por exemplo, passam a ficar expostos aos problemas e vulnerabilidades inerentes a essas redes e podem sofrer interrupções em seus fluxos de dados.

Um outro agravante consiste na seleção da tecnologia e tipo de rede de acesso mais adequado em áreas com inúmeras redes sobrepostas. O *handoff* de uma rede para a outra sem uma avaliação prévia das condições da nova rede pode comprometer seriamente a transmissão de dados dos usuários, além de expô-los a riscos de segurança e à indisponibilidade de conectividade. A disponibilidade de conectividade em redes heterogêneas sem fio é afetada pela *criticidade* dos enlaces de comunicação. Os enlaces são considerados *críticos* a medida que sua interrupção indisponibiliza os serviços de comunicação, tais como o roteamento de pacotes e a comunicação fim-a-fim. Uma variedade de ameaças, como ataques, acidentes e falhas, pode causar degradações menores ou maiores independente do tipo de tecnologia utilizada. Contudo, as vulnerabilidades inicialmente restritas a um determinado cenário podem ser exploradas de modo mais amplo nesses ambientes heterogêneos convergentes.

A resiliência, decorrente da capacidade da rede de proporcionar e manter um nível aceitável do serviço em face a falhas e desafios para sua operação normal tem sido uma estratégia amplamente utilizada no combate às degradações na disponibilidade da conectividade [Heegaard and Trivedi 2009, Sterbenza et al. 2010]. O uso da tolerância a desafios, que consiste na capacidade de um sistema tolerar faltas tal que não ocorram falhas de serviço [Group 2004], também tem sido discutido. Contudo, essa abordagem depende de redundância como uma técnica para compensar falhas aleatórias e não correlacionadas. Logo, a tolerância a desafio não é suficiente diante de múltiplas falhas correlacionadas, portanto é necessária, mas não suficiente para prover resiliência e segurança à rede.

Deste modo, as soluções que garantam resiliência e segurança à rede são fundamentais para tolerar as frequentes desconexões dos dispositivos móveis e ainda assim garantir a disponibilidade de conectividade através de diferentes tecnologias de comunicação [Melo et al. 2013, Lima et al. 2009]. As estratégias que analisam tanto a *fragilidade* como a *robustez* da conectividade da rede são importantes para o desenvolvimento proativo de contramedidas que possam evitar interrupções, ou maiores degradações, nos serviços de comunicação. Além de auxiliarem na análise das condições da rede, permitindo um acesso seguro em redes heterogêneas sem fio sobrepostas.

Este trabalho apresenta um sistema indicador de resiliência na conectividade de redes heterogêneas sem fio. O objetivo do sistema consiste em indicar quais redes são momentaneamente mais resilientes sob o aspecto de disponibilidade de conectividade, o que possibilita uma escolha adequada do acesso em um ambiente sobreposto por inúmeras redes heterogêneas sem fio. Outra característica deste sistema consiste em avaliar e apontar os índices de fragilidade e robustez de conectividade para que estratégias e contramedidas possam ser utilizadas proativamente garantindo resiliência e segurança à rede.

A *fragilidade* da conectividade da rede é verificada pela criticidade dos enlaces de comunicação que correspondem aos links mais vulneráveis a falhas de operação e que podem desconectar a rede a medida que forem afetados. Em contrapartida, a *robustez* é analisada pela existência de rotas redundantes entre vizinhos de um dado dispositivo na rede, que se tornam alternativas as falhas de conectividade em uma dada comunicação. A aferição da fragilidade e da robustez permite determinar qual rede possui melhor condição de conectividade em um dado instante entre as diferentes redes disponíveis. Essa informação possibilita a escolha e a seleção da melhor rede de acesso no momento de transição por áreas com inúmeras redes heterogêneas sem fio sobrepostas.

A avaliação do sistema verifica a eficácia do indicador de resiliência no cálculo da fragilidade e da robustez das conectividades em redes heterogêneas sem fio. Foram utilizados traços de redes heterogêneas reais do projeto *MeshNet*. Os resultados mostraram que o sistema proposto conseguiu indicar com precisão o conjunto de enlaces críticos, bem como as rotas redundantes presentes na rede. Além de permitir seu uso com efetividade para a análise da fragilidade e da robustez de conectividade de redes heterogêneas sem fio, independente do tipo de tecnologia utilizada.

O restante deste trabalho está organizado da seguinte maneira: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a modelagem do sistema. A Seção 4 detalha o sistema proposto, bem como o seu funcionamento. A Seção 5 apresenta a análise e avaliação do sistema e a Seção 6 conclui o trabalho.

2. Trabalhos relacionados

Trabalhos na literatura têm destacado a necessidade de tornar a conectividade das redes heterogêneas sem fio a fim de tolerar as frequentes desconexões de dispositivos móveis e os problemas de indisponibilidade, causados pela transição de terminais que migram suas conexões por redes de diferentes tecnologias de comunicação [Zhang et al. 2008, Ben Hadj Said et al. 2012, Ghosh et al. 2012, Lin et al. 2011]. O trabalho de [Sun et al. 2005] apresenta uma arquitetura para o gerenciamento de conectividade ciente de contexto para dispositivos móveis em redes heterogêneas sem fio. A plataforma fornece interfaces para os aplicativos consultarem a QoS de rede e as condições de disponibilidade da conectividade. Apesar da avaliação e análise da disponibilidade de conectividade, a proposta não considera aspectos de resiliência como estratégia de segurança.

O trabalho de [Gardner et al. 2013] descreve um algoritmo de *Self-Pruning* para a identificação de eventos raros e para a análise de resiliência da rede. O método de análise da resiliência utiliza características da própria rede com base no impacto dos eventos acontecidos. Os autores consideram eventos raros mais importantes na rede, como desastres naturais, rompimento de enlaces, catástrofes e outros, que não são identificados pelas abordagens tradicionais. Uma métrica para mensurar o impacto desses eventos em

uma rede heterogênea sem fio formada por cidades dos EUA foi apresentada. As medidas de conectividade e a capacidade da rede também foram utilizadas na análise. Os resultados mostraram que o impacto de um evento raro pode ser usado como uma métrica de avaliação da rede. Contudo, os próprios autores mencionam a necessidade de avaliação da resiliência da topologia da rede a fim de determinar os índices de fragilidade e robustez de sua conectividade.

O trabalho de [Zhang and Sundaram 2012] apresenta uma avaliação da robustez com base em teoria de redes complexas. Os autores consideram como métrica principal o grau mínimo dos vértices para medida de robustez. Os autores discutem a dificuldade de consenso entre os nós da rede com a presença de agentes mau intencionados difundindo informação. A modelagem analítica apresentada utilizou o modelo de redes complexas de Erdos-Rényi com grafos randômicos de larga escala. Os resultados mostram que o consenso pode ser alcançado de modo resiliente, e sem a necessidade de informação global, em grafos que são suficientemente robustos. No entanto, o uso desta métrica como método de determinar a robustez pode não ser suficiente para outros modelos de redes mais generalizadas como redes heterogêneas que possuem uma alta dinamicidade em sua infraestrutura devido às frequentes associações e dissociações dos dispositivos na rede.

3. Modelo de conectividade da rede

Esta seção descreve as premissas assumidas e prevê a definição dos enlaces críticos para redes heterogêneas sem fio. A conectividade da rede consiste na existência de uma conexão, direta ou através de enlaces intermediários, entre quaisquer dois dispositivos na rede, independente do tipo de tecnologia utilizada. A natureza dinâmica das redes heterogêneas sem fio é resultado da mobilidade dos dispositivos ou do uso de diferentes tecnologias de comunicação. Um *grafo dinâmico não direcionado* $G = (V, E)$, representa a rede heterogêneas sem fio, em que V corresponde a um conjunto finito de vértices que caracterizam os dispositivos (nós) da rede, e E é um conjunto finito de arestas indicando os enlaces (links) entre os pares de dispositivos. Os *grafos dinâmicos* são atualizados pela inserção ou remoção de vértices e arestas a qualquer momento. Assim, dada uma aresta $e = \{u, v\} \in E$, ela é dita incidente em u e $v \in V$. Logo, uma aresta entre u e v representa um enlace na rede com comunicação em ambos os sentidos.

O grafo G' obtido a partir de G representa um instante qualquer t, em que a rede está totalmente conectada. Assim, em cada instante t, existe um grafo completamente conexo para a topologia de conectividade da rede. Um grafo G' é totalmente conexo, se existe um *caminho* P_v^u para quaisquer $u, v \in V$. Logo, um caminho entre dois vértices u e v em G' significa uma sequência de arestas em E que liga uma sequência de vértices em V. Deste modo, G' é chamado *conexo*, se para cada u, v um P_v^u existe. Na rede, um *caminho* compreende uma conexão (rota) entre dois nós ligados diretamente ou através de nós intermédios, independente do tipo de tecnologia de transmissão utilizada. A *distância* $d(u, v)$ entre os dois vértices u e v corresponde ao número de arestas que existe no caminho entre u e v. O *caminho mínimo* P_v^u indica um caminho de *distância mínima* $d(u, v)$. Na rede, a *distância* entre dois nós consiste no número de enlaces existentes na conexão, e o *caminho mínimo* corresponde a uma conexão com menor número de enlaces.

Um enlace crítico consiste em qualquer enlace que se quebrado, por algum evento inesperado, desconecta a rede gerando uma falha de conectividade. Sendo assim, dado

um grafo G' um *corte* $C_{G'}$ representa uma partição dos vértices V em dois subconjuntos disjuntos $\{X, Y\}$, unidos por pelo menos uma aresta. Este corte $C_{G'}$ indica os enlaces mais propícios a falhas de conectividade da rede, causando sua separação em pelo menos duas novas redes distintas. Na rede, as arestas de corte são consideradas enlaces críticos. Portanto, um cut_v^u de G' corresponde a uma divisão nos conjuntos $\{X, Y\}$ em que $u \in X$ e $v \in Y$. O tamanho de $C_{G'}$ consiste no número de enlaces críticos que, se removidos, desconectam a rede. O *corte mínimo (mincut)* de um grafo G' compreende um $C_{G'}$ com o menor número de arestas (caso não ponderada, ou a menor soma de pesos das arestas caso ponderadas), ou seja, o número mínimo de enlaces críticos, que se falhos resultam em uma interrupção da conectividade da rede. A identificação do *corte mínimo* ajuda a apontar enlaces vulneráveis em uma conexão de rede.

Uma *árvore de corte mínimo* T_C de G' corresponde a um grafo induzido pela remoção de arestas, de modo que exista apenas um único caminho, de menor distância, para qualquer dois vértices u e v de G'. Formalmente, T_C consiste em uma árvore tal que, para cada $u, v \in V$, um corte induzido pela remoção do conjunto de arestas mínimas de P_v^u em T_C é um $mincut_v^u$ de G'. Para a rede, T_C significa uma nova rede formada com conexões de menor número de enlaces entre quaisquer dois dispositivos, sem a existência de rotas alternativas entre eles. A descoberta do *corte mínimo* em um caminho específico auxilia na identificação de pontos de vulnerabilidades de enlaces na rede dado um dispositivo origem e um destino.

4. Sistema indicador de resiliência na conectividade de redes heterogêneas

O sistema indicador de resiliência na conectividade das redes heterogêneas sem fio calcula os índices de fragilidade e de robustez e indica quais redes são momentaneamente mais seguras em termos de disponibilidade de conectividade. Isso possibilita a escolha adequada do acesso em um ambiente sobreposto por inúmeras redes heterogêneas sem fio. Além disso, esses indicadores poderão auxiliar na definição de estratégias e contramedidas a serem desenvolvidas a fim de prover princípios de resiliência e segurança.

O sistema compreende três componentes: o *Verificador de Conectividade*, o *Medidor de Criticidade de Enlaces* e o *Medidor de Redundância de Rotas*, conforme ilustrado na Figura 1. O sistema recebe como entrada informações da topologia de cada rede de acesso detectada, extrai um grafo de conectividade, calcula a criticidade dos enlaces e a redundância de rotas em cada rede, e fornece um índice de fragilidade e robustez da conectividade de cada uma das redes disponíveis.

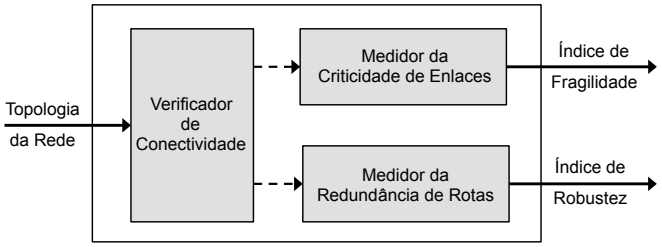


Figura 1. Arquitetura do sistema indicador de resiliência

O componente *Verificador de Conectividade* extrai um grafo de conectividade da rede a partir das informações de sua topologia. Na sequência, verifica a existência de

caminhos para todos os nós do grafo, constatando sua conectividade. Se nós isolados forem detectados classifica-se a rede como desconexa, caso contrário, sua conectividade será avaliada. O grafo de conectividade é encaminhado para os componentes medidor de criticidade de enlaces e medidor de redundância de rotas.

O componente *Medidor de Criticidade de Enlaces* recebe o grafo de conectividade e calcula os enlaces críticos, isto é, aqueles que se removidos são capazes de desconectá-lo. Esses enlaces são referência para o cálculo do índice de *Fragilidade de Conectividade da Rede* ou apenas *Fragilidade da Rede* (NF). O *Medidor de Redundância de Rotas* utiliza o grafo de conectividade da rede para verificar a existência de enlaces entres os vizinhos de um nó. Esse processo se repete até que a análise seja feita para todos os nós do grafo. O resultado desta análise é utilizado para caracterizar o índice de *Robustez de Conectividade da Rede* ou simplesmente *Robustez da Rede* (NR). As etapas para o cálculo destes dois índices são detalhadas a seguir.

4.1. Medidor de criticidade de enlaces

A fragilidade da rede está diretamente relacionada com a vulnerabilidade a falha dos enlaces críticos. Desta forma, o cálculo da fragilidade tem como referência as *árvores de corte mínimo*, as quais indicam em um grafo a quantidade de enlaces necessários para desconectar a rede. As árvores de corte mínimo contém todos os vértices do grafo de topologia de conectividade da rede e um conjunto de arestas ponderadas. O peso de cada aresta na árvore corresponde ao número de enlaces necessários para a desconexão.

Dado um grafo G' que representa a topologia de conectividade da rede em um instante t, o algoritmo de árvores de corte mínimo retorna um novo grafo ponderado T_C. O w representa o peso de cada aresta e W_c o conjunto de todos os w do grafo. Os pesos W_c correspondem a todos os cortes mínimos $mincut_v^u$ entre todos os pares de vértices u, v. O $mincut_v^u$ representa o menor número de enlaces entre u, v que, se removidos, desconectam o grafo. Estes por sua vez são denominados enlaces críticos.

A *fragilidade da rede* é calculada a partir da árvore de corte mínimo T_C e do conjunto de pesos W_c. Com base nos resultados apresentados em [J. and M'Raïhi 1998] [Arce 2003], a fragilidade da rede é obtida por uma relação entre o menor e o maior número de enlaces expostos a perturbações na topologia de conectividade da rede. Logo, NF indica a relação entre o peso mínimo e o máximo de T_C para todos os pesos w pertencentes à W_c, como mostrado pela Equação 1. Uma topologia com *alta fragilidade* corresponde àquela que necessita de um menor número de enlaces removidos para desconectá-la, caso contrário, define-se como topologia de *baixa fragilidade*.

$$NF = \frac{min\{w \mid \forall\, w \in W_c\}}{max\{w \mid \forall\, w \in W_c\}} \tag{1}$$

4.2. Medidor de redundância de rotas

A existência de rotas redundantes representa a robustez de conectividade da rede. Essas rotas se tornam alternativas às falhas de conectividade em uma dada comunicação, particularmente entre os vértices críticos (CV), que são conectados por enlaces críticos. A robustez é aferida por meio da técnica de agrupamento (clusterização), que verifica as conexões existente entre os vizinhos dos vértices no grafo. A partir desta medida, calcula-se

um índice local, para um único nó e um índice global para a rede como um todo, que serão usados na definição de NR.

Dado um grafo G', o vértice v é dito vizinho de u se existe uma aresta direta entre ambos. O grau de v corresponde à soma de seus vizinhos, denotado por d_v. O coeficiente de agrupamento de v consiste na quantidade de arestas que os vizinhos de v têm entre eles, dividido pela quantidade total de arestas que v poderia ter. A partir de d_v, o maior número de arestas que v pode ter é dado por $B = \binom{d_v}{2}$. Seja E_v o número real de arestas que v possui, ou seja, o seu número atual de vizinhos, o coeficiente de agrupamento de v é definido pela Equação 2. O coeficiente de agrupamento C_v indica o nível de redundância que um nó tem em termos de conexões. Esta medida também indica o número de cliques, de tamanho 3, no grafo de conectividade da rede.

$$C_v = \frac{E_v}{B} = \frac{2 \cdot E_v}{d_v \cdot (d_v - 1)} \tag{2}$$

O coeficiente de agrupamento local C_v dos vértices possibilita calcular um valor global para a rede toda, C_{Global}. Neste caso, definido como o menor C_v entre todos os vértices de G', como mostrado na Equação 3.

$$C_{Global} = min\{C_v \mid \forall\ v \in V_G\} \tag{3}$$

Os valores de C_{Global} e C_v permitem calcular o valor da NR, denotado pela Equação 4. Esta medida incide sobre os enlaces e os nós mais frágeis da rede. Logo, NR implica na relação entre o C_{Global} e o máximo C_v calculado para todos os vértices contidos no conjunto de vértices críticos (CV). Por sua vez, CV corresponde a todos os vértices presentes no caminho relacionado ao menor corte mínimo $mincut_v^u$ de T_C.

$$NR = \frac{C_{Global}}{max\{C_v \mid \forall\ v \in CV\}} \tag{4}$$

Com os índices de fragilidade e robustez fornecidos pelo sistema indicador de segurança cada dispositivo móvel em transição por diferentes redes de acesso consegue selecionar a rede mais segura em termos de disponibilidade de conectividade em ambientes sobrepostos por inúmeras redes heterogêneas sem fio. Além de uma avaliação momentânea das condições de conectividade da rede o sistema ajuda a evitar que os dispositivos móveis em transição se conectem em redes de acesso comprometidas e tenha suas comunicações interrompidas. Outra vantagem dos índices inferidos pelo sistema consiste em permitir que medidas proativas de prevenção possam ser desenvolvidas para evitar a indisponibilidade de conectividade da rede garantindo resiliência e segurança.

4.3. Funcionamento do sistema indicador de resiliência

Para exemplificar o funcionamento do sistema indicador de resiliência foi considerado um cenário em que um usuário portador de um dispositivo computacional móvel encontra-se em uma área de sobreposição de duas (ou mais) redes de acesso heterogêneas sem fio, como ilustrado na Figura 2. Antes de se conectar em qualquer uma das redes o sistema

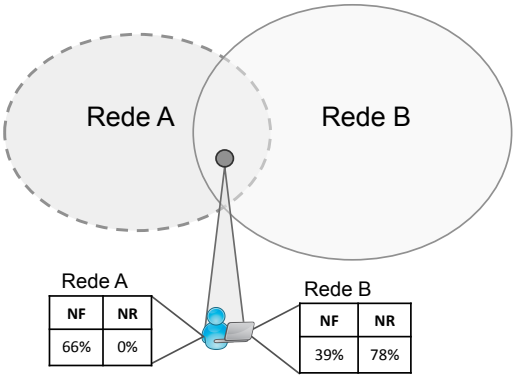


Figura 2. Cenário de funcionamento do sistema indicador de resiliência

avalia as condições de conectividade indicando os índices de fragilidade e robustez de ambas as rede. Com esses indicadores o usuário pode se conectar na rede mais resiliente.

O sistema indicador inicia sua operação a partir da obtenção de informações da topologia de todas as redes detectadas. Neste exemplo, as informações da topologia são representadas por uma *lista de adjacência* disponibilizada pelo provedor de acesso de cada rede. O componente *Verificador de Conectividade* realiza uma *busca em largura* nos dados da topologia da rede e retorna um grafo de conectividade da mesma. A Figura 3(a) ilustra o grafo de conectividade da "Rede A".

O componente *Medidor de Criticidade de Enlaces* recebe o grafo de conectividade e identifica o conjunto de enlaces críticos através da árvore de corte mínimo. Neste trabalho, a árvore de corte mínimo foi obtida por meio do algoritmo de Gomory-Hu [Gomory and Hu 1961]. O algoritmo de Gomory-Hu tem como entrada um grafo da topologia de conectividade e retorna uma árvore de corte mínimo. A árvore de corte mínimo T_C e o conjunto de pesos W_c, obtidos com o algoritmo são construídos com a computação de $|V| - 1$ corte mínimo.

A Figura 3(a) mostra um grafo G' que representa uma topologia de conectividade da "Rede A" em um instante t. A Figura 3(b) ilustra uma árvore de corte mínimo T_C extraído do grafo G' e seus pesos w representam a quantidade de arestas, que são necessárias para desconectar G'. Por exemplo, em T_C ilustrada pela Figura 3(b), a aresta tracejada em destaque entre os vértices 4 e 3 possui um peso w de valor 2, o que significa que se duas arestas que conectam os vértices 4 e 3 forem removidas, a rede será desconectada. De mesmo modo, diferentes arestas em T_C empregam essa mesma lógica.

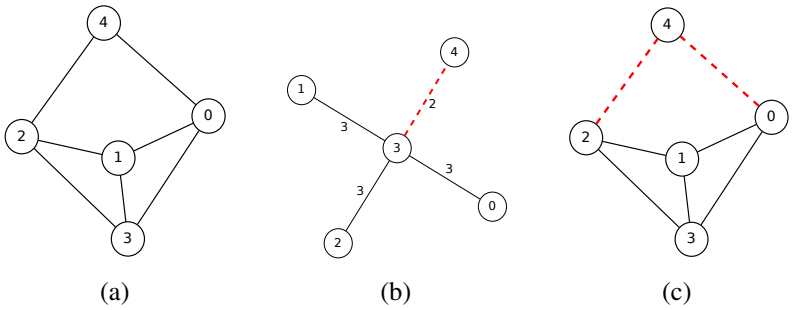
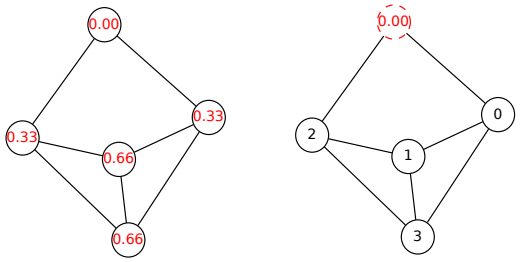


Figura 3. Análise de fragilidade

Na Figura 3(b), entre todas as arestas de T_C, o peso w mínimo corresponde 2 e o máximo 3, para todo conjunto W_c. Assim, usando a Equação 1, o resultado é de aproximadamente 0,66, o que indica que o nível de fragilidade para esta rede corresponde a 66% (em relação aos seus enlaces). Com base neste exemplo, o corte mínimo envolve dois enlaces diferentes na topologia da conectividade da rede. Estes dois enlaces são chamados enlaces críticos e estão nas conexões entre os nós 4 e 3, como mostrado na Figura 3(c) pelas linhas tracejadas. Logo, os vértices conectados por enlaces críticos são denominados como vértices críticos. Na Figura 3(c), os vértices críticos são 4, 2 e 0.

Em paralelo à operação realizada pelo componente *Medidor de Criticidade de Enlaces*, o terceiro componente do sistema, o *Medidor de Redundância de Rotas* também é executado. Esse componente calcula o coeficiente de agrupamento dos vertices do grafo de conectividade. A Figura 4(a) mostra o coeficiente de agrupamento local C_v calculado para cada vértice e a Figura 4(b) mostra o valor do coeficiente de agrupamento global C_{Global} calculado para toda a rede. Dado que o valor de C_{global} implica no menor C_v, calculado para todo conjunto de vértice V do grafo G', o valor de C_{global} para essa rede exemplificada pela Figura 4(a) corresponde a 0,0. Deste modo, utilizando a Equação 4, para esse exemplo, tem-se que C_{Global} corresponde a 0,0 e o maior C_v equivale a 0,66. Logo, o valor de NR implica em 0,0, e indica o índice de robustez da rede.



(a) (b)

Figura 4. Análise de robustez

Esse procedimento realizado para a análise das condições de conectividade da "Rede A" também é realizado para a "Rede B". Assim, a partir dos resultados dos índices de fragilidade e robustez de cada rede o dispositivo móvel em transição poderá se conectar na rede de acesso mais segura em termos de disponibilidade de conectividade.

5. Análise e avaliação do sistema indicador de resiliência

Esta seção apresenta a análise do sistema indicador de resiliência na conectividade de redes heterogêneas sem fio, cujo o objetivo consiste em identificar os índices de fragilidade e robustez da conectividade dessas redes.

5.1. Metodologia

A análise do sistema indicador de resiliência foi realizada utilizando traços reais da rede *mesh* do projeto *UCSB MeshNet* da Universidade da Califórnia em Santa Barbara, nos Estados Unidos. A rede representa um ambiente de um campus universitário, composta por 19 nós que operam nos padrões *802.11a/b*, criando uma rede heterogênea em termos de seus padrões de comunicação [Meshnet 2013]. Os dispositivos móveis conectados na

rede fazem transições de suas conexões durante a sua mobilidade, mudando constantemente a topologia de conectividade da rede. Os dados utilizados correspondem a testes realizados em 2007 e estão disponíveis no repositório web CRAWDAD[1].

A base de dados é formada por 900 arquivos contendo as listas de adjacências da topologia da rede. Cada arquivo representa um instante de tempo com as condições de conectividade da rede. Com o objetivo de avaliar apenas os instantes em que a rede era conexa, um script subtrai os arquivos que continham instantes em que a rede era desconexa, resultando em 577 arquivos. Cada linha dos arquivos determina as conexões entre os nós. O primeiro elemento da linha consiste no endereço IP de um nó específico, seguido de uma lista que contém o endereço IP com quem o primeiro nó tem conexão. Um exemplo pode ser visualizado no quadro a seguir.

10.1.1.2	10.1.1.60	10.1.1.9	10.1.1.100	10.1.1.103	10.1.1.5

Essa linha representa as conexões do nó 10.1.1.2. Nesse instante de tempo, ele possui conexões com os nós 10.1.1.60, 10.1.1.9, 10.1.1.100, 10.1.1.103 e 10.1.1.5. Para a representação da conectividade da rede a Figura 5 ilustra seis topologias da mesma rede em diferentes instantes de tempo t.

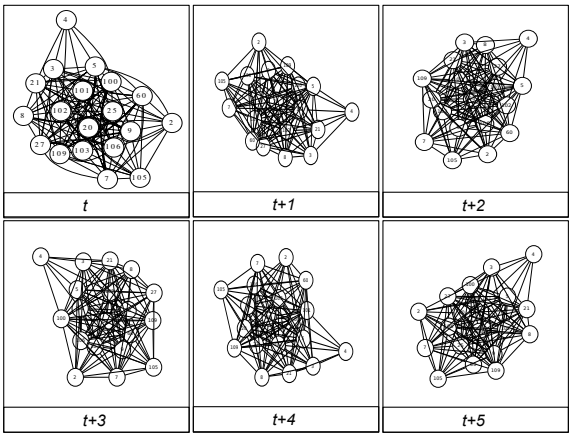


Figura 5. Exemplos de grafos de conectividade da rede em diferentes instantes

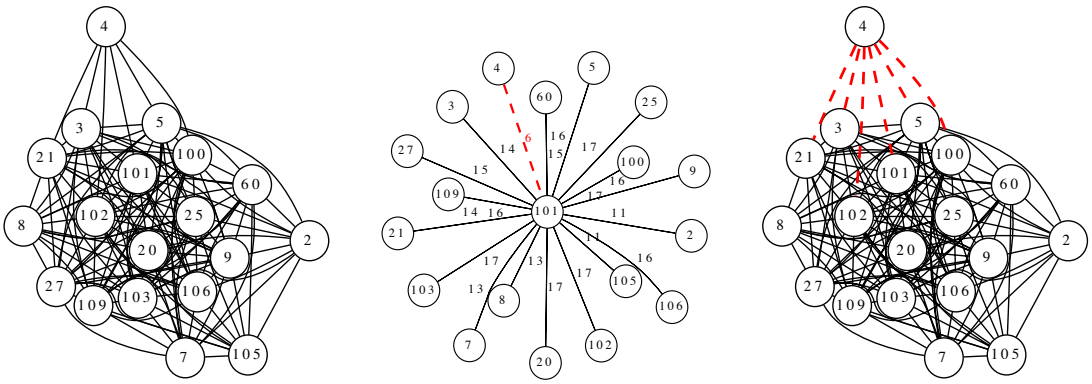
A partir do grafo foi possível definir uma estrutura de dados para a execução do algoritmo de árvores de corte mínimos de *Gomory-Hu* e para o cálculo do coeficiente de agrupamento. A biblioteca LEMON fornece diversos algoritmos e estruturas de dados próprias para a manipulação de dados relacionados a teoria de grafos [Lemon 2013]. Neste trabalho, a biblioteca LEMON foi utilizada para extrair a árvore de corte mínimo de cada topologia utilizada. Um script Python foi implementado para automatizar todo o processo de análise dos vários arquivos e grafos. O script faz a leitura e o tratamento dos dados, depois com o grafo estruturado é gerada uma segunda estrutura específica da biblioteca LEMON e nessa etapa as informações necessárias são coletadas para o cálculo do coeficiente de agrupamento individual. Na etapa seguinte o algoritmo de *Gomory-Hu* é executado e o coeficiente de agrupamento é calculado, a partir do algoritmo, os $mincut_v^u$ de cada vértice são calculados. Na última etapa o script organiza esses dados em uma tabela para análise futura.

[1]http://crawdad.cs.dartmouth.edu/meta.php?name=ucsb/meshnet

5.2. Avaliação da fragilidade

A *fragilidade alta* consiste na menor quantidade de enlaces necessária para a desconexão da rede. Já a *fragilidade baixa* consiste na maior quantidade de enlaces críticos para a desconexão. Assim, quanto menos enlaces necessários para desconectar a rede maior sua fragilidade, quanto mais enlaces necessários para sua desconexão menor é a fragilidade da rede. O algoritmo de árvore de corte mínimo de *Gomory-Hu* foi utilizado para quantificar o conjunto de enlaces necessários para a desconexão da rede, bem como, identificar a localização desses enlaces, considerados críticos.

A Figura 6(a) ilustra um grafo G' representando um determinado instante t de conectividade da rede *MeshNet*, ao aplicar o algoritmo de *Gomory-Hu* obtemos a árvore de corte mínimo, Figura 6(b), e partir desta árvore, podemos identificar que a quantidade mínima de arestas necessárias para a desconexão do grafo é seis (indicada pelo peso da linha pontilhada), as quais conectam o nó 4 ao restante da rede. A partir da quantidade de arestas descobertas que desconectam a rede, realizamos a identificação e a localização destas arestas no grafo original, indicadas pelas linhas tracejadas na Figura 6(c). Como visto na Figura 6(c), as arestas que conectam o nó 4 aos nós $3, 5, 21, 100, 101$ e 102 são os enlaces críticos da rede no instante t. Deste modo, ao remover apenas esse conjunto mínimo de enlaces, desconecta-se a rede.



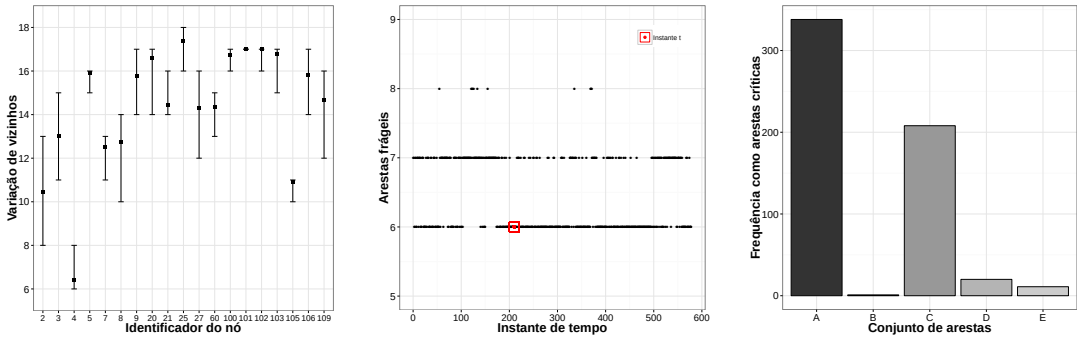
(a) (b) (c)

Figura 6. Análise do grafos de conectividade no instante t

Posterior à análise de um instante específico, foi realizado o processamento de todos os arquivos representando as condições global da rede para todo seu período de funcionamento. A Figura 7(a) ilustra a variação da quantidade de vizinhos de cada nó em todos os instantes da rede, o que mostra a dinamicidade durante seu funcionamento. A Figura 7(b) apresenta a variação do número de arestas críticas da rede.

Apesar da dificuldade em garantir a conectividade a todo instante, observa-se certa estabilidade entre os valores de enlaces críticos, variando entre seis e oito arestas do grafo de conectividade. Outro comportamento observado é a existência de repetições de alguns enlaces específicos, que aparecem no grupo de arestas de fragilidade alta. Isso indica que esses enlaces constantemente estão entre os mais frágeis, como ilustrado na Figura 7(c).

A Tabela 1 ilustra as arestas pertencentes a cada conjunto de enlaces críticos. A Figura 7(c) mostra que as arestas pertencentes ao conjunto A aparecem constantemente



(a) (b) (c)

Figura 7. Avaliação da fragilidade

entre as arestas de fragilidade alta. Deste modo, é possível afirmar que as arestas do conjunto A são os enlaces mais frágeis da rede.

Conjuntos	Arestas							
A	4, 101	3, 4	4, 100	4, 5	4, 21	4, 102	-	-
B	4, 101	3, 4	4, 100	4, 5	4, 21	4, 25	-	-
C	4, 101	3, 4	4, 100	4, 5	4, 21	4, 25	4, 102	-
D	4, 101	4, 8	3, 4	4, 100	4, 5	4, 21	4, 102	-
E	4, 101	4, 8	3, 4	4, 100	4, 5	4, 21	4, 25	4, 102

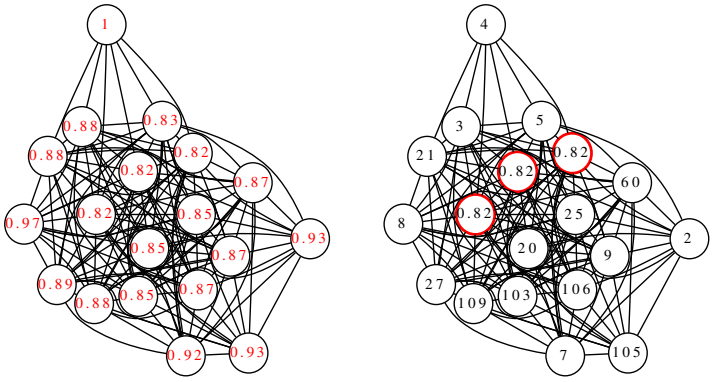
Tabela 1. Descrição do conjunto de arestas da Figura7(c).

5.3. Avaliação da robustez

A robustez foi mensurada de forma local pelo número de conexões entre os vizinhos de um nó, e de forma global para toda a rede. Assim, a robustez da conectividade da rede é determinada pela relação entre o índice global e o índice local, deste modo, nenhum nó frágil será ocultado pela média do coeficiente de agrupamento. Dado um determinado instante t de conectividade da rede ilustrado pela Figura 6(a), foi aplicada a técnica de clusterização para encontrar os valores dos coeficientes de agrupamento locais, como observado na Figura 8(a), em que os valores de cada nó representam seu índice de robustez. Como o agrupamento global da rede é representada pelo menor índice local, a Figura 8(b) ilustra o grafo do instante t com o nó que determina o valor do coeficiente de agrupamento global da rede. Como visto na Figura 8(b), três nós possuem os mesmos valores, correspondentes ao menor índice da rede. Logo, qualquer um deles pode ser utilizado para o cálculo do índice de robustez da rede.

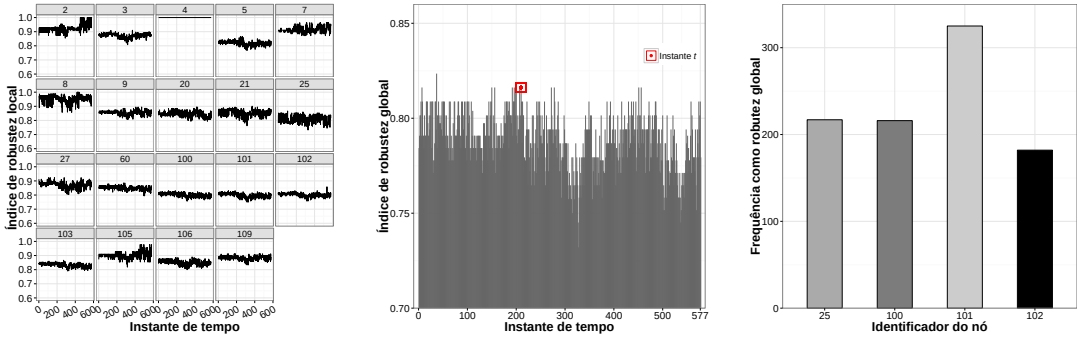
Após a análise individual do instante t, foram realizadas as análises para todo o período de funcionamento da rede a fim de encontrar a variação dos valores de clusterização local de cada nó e os valores para a clusterização global. A Figura 9(a) mostra os gráficos de variação da clusterização local de cada nó durante todos os 577 instantes em que a rede esteve totalmente conectada. A Figura 9(b) ilustra a variação da clusterização global durante todo tempo de atividade da rede.

Com base nos resultados é possível identificar qual o nó que por maior período determina o valor de agrupamento global da rede, e define o valor de robustez da mesma, por ser o nó de menor índice de clusterização. A Figura 9(c) ilustra o número de vezes que um nó determina o valor da clusterização global da rede. Logo, a partir dos resultados é possível afirmar que o nó 101 corresponde ao nó que por maior período determinou o



(a) (b)

Figura 8. Análise de agrupamento da rede



(a) (b) (c)

Figura 9. Avaliação da robustez

valor de clusterização global. Deste modo, esse mesmo nó foi usado por mais vezes para o cálculo do valor de robustez da rede.

6. Conclusão

O trabalho apresentou um sistema que mede a resiliência de redes heterogêneas sem fio no momento da tomada de decisão de transição de acesso entre redes. Este sistema indica quais redes são momentaneamente mais seguras sob o aspecto de disponibilidade de conectividade, possibilitando aos dispositivos móveis em transição uma escolha adequada do acesso em um ambiente sobreposto por inúmeras redes heterogêneas sem fio. O sistema avalia e aponta os índices de fragilidade e robustez de conectividade de modo que estratégias e contramedidas possam ser aplicadas para garantir a resiliência e segurança à rede. A avaliação do sistema considera traços reais de uma rede heterogênea sem fio. Os resultados mostraram a eficácia do sistema na identificação dos índices de fragilidade e de robustez da conectividade da rede em diferentes momentos de avaliação. Como trabalho futuro, pretende-se usar esse sistema para apoiar o desenvolvimento de soluções voltadas à resiliência e a segurança de redes heterogêneas sem fio.

Referências

Arce, I. (2003). The weakest link revisited. *IEEE Security Privacy*, 1(2):72–76.

Ben Hadj Said, S., Guillouard, K., and Bonnin, J. (2012). On the need for adaptive connectivity management in multi-access architectures. In *IEEE NOF*, páginas 1–5.

Gardner, M., Beard, C., and Medhi, D. (2013). Using network measure to reduce state space enumeration in resilient networks. In *IEEE DRCN*, páginas 250–257.

Ghosh, A., Mangalvedhe, N., Ratasuk, R., Mondal, B., Cudak, M., Visotsky, E., Thomas, T. A., Andrews, J. G., Xia, P., Jo, H. S., Dhillon, H. S., and Novlan, T. D. (2012). Heterogeneous cellular networks: From theory to practice. *IEEE Communications Magazine*, 50(6):54 –64.

Gomory, R. E. and Hu, T. C. (1961). Multi-Terminal Network Flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570.

Group, T. W. (2004). Reliability-related metrics and terminology for network elements in evolving communications networks. *Alliance for Telecommunications Industry Solutions (ATIS)*.

He, D., Bu, J., and Zhang, Y. (2013). Security and efficiency in roaming services for wireless networks: challenges, approaches, and prospects. *IEEE Communications Magazine*.

Heegaard, P. E. and Trivedi, K. S. (2009). Network survivability modeling. *Computer Networks*, 53(8):1215–1234.

J., M. and M'Raïhi, D. (1998). Mix-based electronic payments. In *Proc. of the Selected Areas in Cryptography*, páginas 157–173.

Lemon (2013). Library for efficient modeling and optimization in networks. `http://lemon.cs.elte.hu/trac/lemon`. Último acesso: junho 2014.

Lima, M., dos Santos, A. L., and Pujolle, G. (2009). A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 11(1):66–77.

Lin, P., Zhang, J., Chen, Y., and Zhang, Q. (2011). Macro-femto heterogeneous network deployment and management: from business models to technical solutions. *IEEE Wireless Communications*, páginas 64–70.

Melo, R., Santos, A., , Nogueira, M., and Medhi, D. (2013). *Anais / 31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, chapter Modelagem e projeto de redes sem fio heterogêneas, resilientes e sobreviventes, páginas 1–50. SBC.

Meshnet (2013). Mesh testbed is an experimental wireless mesh network. `http://moment.cs.ucsb.edu/meshnet/`. Último acesso: Junho 2014.

Sterbenza, J. P. G., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Scholler, M., and Smith, P. (2010). Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 58(1):1245–1265.

Sun, J.-Z., Riekki, J., Jurmu, M., and Sauvola, J. (2005). Adaptive connectivity management middleware for heterogeneous wireless networks. *IEEE Wireless Communications*, 12(6).

Zhang, C., Song, Y., and Fang, Y. (2008). Modeling secure connectivity of self-organized wireless ad hoc networks. In *IEEE INFOCOM - The 27th Conference on Computer Communications*.

Zhang, H. and Sundaram, S. (2012). Robustness of complex networks with implications for consensus and contagion. In *IEEE CDC*, páginas 3426–3432.

Um Sistema de Detecção de Ataques Sinkhole sobre 6LoWPAN para Internet das Coisas

Christian Cervantes, Diego Poplade, Michele Nogueira, Aldri Santos

[1]Núcleo de Redes Sem Fio e Redes Avançadas (NR2)
Universidade Federal do Paraná – Curitiba – Brasil

{cavcervantes,dap10,michele,aldri}@inf.ufpr.br

Abstract. *The networks of Internet of Things (IoT) are formed by heterogeneous objects and these objects have in general very limited resources. Thus, IoT networks are vulnerable to various attacks, being the attack sinkhole one of the most destructive. However, existing solutions to provide protection and security against attacks sinkhole on IoT have high consumption of resources and employ complex mechanisms to ensure good performance. This paper proposes a system, called INTI (intrusion detection of sinkhole attacks on 6LoWPAN for IoT), to identify the presence of sinkhole attacks on the routing service of IoT. INTI aims to prevent, detect and isolate sinkhole attacks on routing within the IoT, while mitigating adverse effects. The system combines the use of watchdog, reputation and trust for detection of attackers, by analyzing the behavior of devices. Simulation results show the INTI performance and efficiency in terms of attack detection rate, number of false positives and false negatives.*

Resumo. *As redes de Internet das coisas (IoT) são formadas por objetos heterogêneos e muito desses objetos possuem recursos limitados. Logo, as redes IoT são vulneráveis a vários tipos de ataques, sendo o ataque sinkhole um dos mais destrutivos. Contudo, as soluções existentes para a proteção e segurança contra os ataques sinkhole em IoT geram um elevado consumo de recursos e usam mecanismos complexos para garantir um bom desempenho. Este trabalho propõe um sistema para identificar ataques sinkhole no serviço de roteamento da IoT, chamado de INTI (Sistema de detecção de* ***I****ntrução de ataques Si****N****khole sobre 6LoWPAN para a Interne****T*** *das Co****I****sas). O INTI visa prevenir, detectar e isolar os ataques sinkhole no roteamento dentro da IoT, e ao mesmo tempo mitigar os efeitos adversos. O INTI combina o uso de watchdog, reputação e confiança para a detecção dos atacantes, por meio da análise do comportamento dos dispositivos. Resultados mostram o desempenho e a eficiência do INTI na detecção de ataques, número de falsos positivos e negativos.*

1. Introdução

Devido aos avanços das tecnologias e a redução dos dispositivos computacionais, estes se tornaram mais acessíveis e mais disponíveis ao público em geral. Baseado nesses avanços, surgiu o conceito da Internet das coisas (IoT). A IoT é uma rede híbrida, aberta e heterogênea que integra dispositivos inteligentes chamados de coisas (*things*), como eletrodomésticos, livros, canetas e carros, entre outros objetos que usualmente não pertencem à computação interatuando com computadores, sensores, celulares, PDAs e outros dispositivos. Estes dispositivos buscam compartilhar informações, dados e recursos, agindo e

reagindo diante de situações e mudanças no ambiente. O objetivo da IoT é possibilitar a integração e a unificação de todos os objetos e sistemas de comunicação que nos cercam.

Com o aumento dos dispositivos inteligentes e a mobilidade de alguns destes, a IoT está exposta a diversas vulnerabilidades na comunicação por apresentar uma infraestrutura variável e a maior parte dos dispositivos possuírem recursos computacionais limitados, como baixa energia, limitada capacidade de processamento, armazenamento, conexão através de links com perdas e outras características [Atzori et al. 2010]. Em razão das características, a IoT torna-se vulnerável a diversas formas de ataques de roteamento [Wallgren et al. 2013]. Dentre esses tipos de ataques na IoT, destaca-se o ataque *sinkhole*, sendo considerado um dos ataques de roteamento mais destrutivos para as redes sem fio [Jin Qi 2012, Bannack et al. 2008]. Um dispositivo atacante *sinkhole* tem o objetivo de atrair a maior quantidade de tráfego de uma certa área prejudicando um ponto de coleta de receber os dados enviados pelos nós.

Apesar de existirem vários trabalhos na literatura que quantificam o impacto do ataque *sinkhole* sobre redes como redes móveis Ad hoc (MANETs), redes de sensores sem fio (RSSFs) e redes veiculares Ad hoc (VANETs) [Sedjelmaci and Feham 2011, Sheela and Mahadevan. 2011, Shafiei et al. 2014, Lima et al. 2009], estas soluções geram outros problemas para a rede denominados de *efeitos adversos*, como elevadas taxas de falsos positivos e falsos negativos, elevado consumo de energia, baixo desempenho do sistema, entre outros. Além disso, poucas pesquisas têm sido desenvolvidas para a proteção e a segurança da IoT na transmissão de informação [Raza et al. 2013, Kasinathan et al. 2013], e estes trabalhos são inadequados para um funcionamento dinâmico porque não consideram a mobilidade dos dispositivos, sendo isso fundamental para seu uso por pessoas e objetos.

Os sistemas de detecção de intrusão (IDS) têm como objetivo melhorar a segurança diante de ataques e ameaças aos sistemas computacionais ou redes de computadores. Alguns trabalhos utilizam estratégias de *watchdog* para a detecção local de nós atacantes [Keally et al. 2010, Wahab et al. 2014], sendo capaz de escutar, analisar os pacotes transmitidos pelo próximo salto (*next-hop*) e dos nós vizinhos. Esta estratégia minimiza o número de falsos positivos e negativos aumentando a eficiência e evita o descarte de pacotes por nós vizinhos. Outros trabalhos usam mecanismos de reputação e confiança [Perez-Toro et al. 2010, Ganeriwal et al. 2008]. Entre as suas vantagens, estão que eles são dinâmicos, temporais e precisam ser atualizados de forma constante para que permitam identificar a origem da ameaça. Este mecanismo, além de ser eficaz, ajuda a reduzir o impacto de ataques na rede conseguindo um bom desempenho do sistema. Contudo, essas estratégias não têm sido aplicadas na IoT embora sejam adequadas.

Este trabalho propõe um sistema para identificar a presença de ataques *sinkhole* dentro do serviço de roteamento da IoT, chamado de INTI (*Sistema de detecção de **I**ntrução de ataques Si**N**khole sobre 6LoWPAN para a Interne**T** das Co**I**sas*). O INTI visa prevenir, detectar e isolar os efeitos do ataque *sinkhole* no serviço de roteamento, e ao mesmo tempo mitigar os *efeitos adversos*. O sistema combina o uso de *watchdog* com o uso reputação e confiança, para a detecção de ataques *sinkhole* na IoT, por meio da análise do comportamento de cada nó. Os resultados da simulação mostram que o INTI garante uma taxa de detecção de pelo menos 92% em um cenário com nós fixo e de 75% em um cenário com nós móveis.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha o modelo da IoT. A Seção 4 descreve o sistema INTI. A Seção 5 apresenta a avaliação e os resultados obtidos pelo sistema diante de ataques. A Seção 6 finaliza o trabalho com as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Diversas técnicas e mecanismos que tratam da segurança na comunicação têm sido utilizados para a detecção de ataques *sinkhole*. Os autores [Moon and Cho. 2009] propõem um IDS utilizando lógica *fuzzy* para a detecção da ataques *sinkhole* em uma RSSFs. Esta abordagem emprega nós mestres a fim de monitorar a comunicação. Os nós atacantes são detectados a partir dos dados coletados pelos nós mestres. Os autores [Sedjelmaci and Feham 2011] propuseram um IDS híbrido baseado em agrupamentos para uma RSSF. Este sistema usa uma combinação entre a detecção de anomalias baseado em máquina de vetor de suporte (SVM) e a detecção através de assinaturas. Ele possui dois módulos: a detecção híbrida (HIDM) e a detecção cooperativa (CDM) dos nós. No HIDM, ocorre o processo de treinamento, onde cada agente IDS treina o SVM localmente. Já o CDM executa um mecanismo de votação para a detecção de nós suspeitos.

O IDS proposto em [Sheela and Mahadevan. 2011] apresenta o uso de agentes móveis para a detecção de ataques *sinkhole*. Estes agentes utilizam dois algoritmos: o algoritmo de navegação e o algoritmo de roteamento. O algoritmo de navegação onde um agente móvel fornece informações da rede quando visita cada nó. O algoritmo de roteamento de dados descreve como um nó usa as informações da rede para rotear os pacotes de dados para não acreditar em caminhos falsos. Os autores [Shafiei et al. 2014] propõem duas abordagem para detectar ataques *sinkhole* em uma RSSF. A lógica utilizada é que os nós ao redor do *sinkhole* esgotam sua energia mais rápido. A primeira abordagem utiliza um método de geoestatística para avaliar a energia residual de cada região. A segunda abordagem possui um método de monitoramento distribuído para detectar regiões com o menor nível de energia residual a fim de detectar o atacante. Contudo, os sistemas abordados possuem a desvantagem de gerar elevadas taxas de falsos positivos e negativos, aumentado o consumo de memória e energia, entre outros. Além disso, essas abordagens não são apropriadas para a IoT por serem muito complexas. Desta forma, faz-se necessário a construção de IDS para IoT baseada em técnicas mais simples e eficazes.

O mecanismo RDAS [Perez-Toro et al. 2010] usa uma abordagem baseada na reputação para identificar e isolar nós maliciosos em uma RSSF. Ele considera a formação de agrupamentos dos nós, onde o líder analisa os dados coletados dos nós do agrupamento para determinar a localização de um evento malicioso, usando a redundância dos dados. Do mesmo modo, o modelo em [Wahab et al. 2014] propõe uma solução para detectar nós egoístas em (VANETs). O modelo consiste em duas fases: (i) motivar os nós para que atuem de maneira cooperativa utilizando incentivos; (ii) o uso de *watchdog* para detectar nós egoístas baseado em evidências cooperativas, aumentando a probabilidade de detecção. Contudo, estas estratégias além de ser eficazes e efetivas, devem ser usadas em conjunto para garantir um ambiente de comunicação segura para a IoT.

Existem na literatura poucas pesquisas desenvolvidas para a proteção e a segurança da IoT na transmissão de informação. O sistema SVELTE [Raza et al. 2013] considera consultas realizadas a partir de um roteador de borda, que percorre todos os

nós de uma rede IoT, a fim de detectar inconsistências. Essas inconsistências são obtidas através de comparações das informações das posições de cada nó. Já o sistema Ebbits [Kasinathan et al. 2013] emprega um componente que escuta o tráfego da rede a fim de realizar uma análise e detectar nós com mau comportamento em redes 6LoWPAN. Apesar destes IDSs atenderem a maioria das características da IoT, eles não consideram a mobilidade e são muitos restritos na análise do comportamento dos nós. Além disso, esses sistemas possuem elevadas taxas de consumo de recursos e baixo desempenho.

3. Modelo da IoT para o Sistema INTI

Esta seção descreve a rede física, o modelo de comunicação e o ataque. Esta rede considera dispositivos heterogêneos, sendo alguns deles móveis. Além disso, a rede possui duas hierarquias: a hierarquia principal e auxiliar. A **hierarquia principal** é a estrutura que permitirá a comunicação entre os diferentes agrupamentos, nesta hierarquia só intervém os nós líderes, os nós associados e a estação-base como alvo. A **hierarquia auxiliar** compreende a comunicação de cada agrupamento realizado pelo nó líder e seus nós membros. Portanto, a vantagem destas hierarquias é que elas permitem a comunicação de várias sub-redes, oferecendo um ganho expressivo na escalabilidade, estabilidade como mostra a Figura 1. A IoT no sistema INTI possui três características: o modelo físico da rede, o modelo de comunicação e o modelo do ataque, como apresentado na Figura 1.

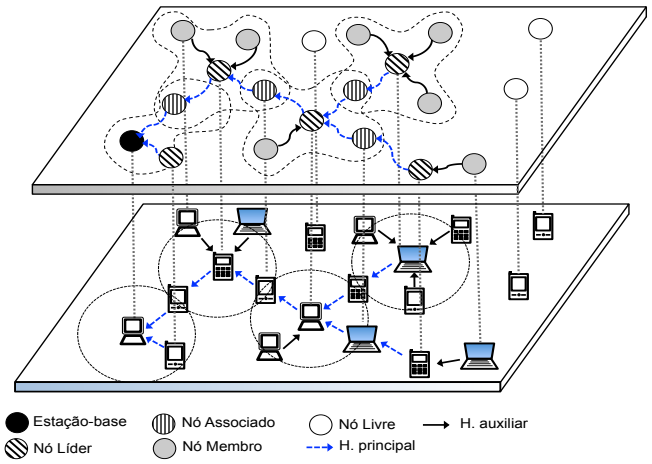


Figura 1. Modelo da IoT

Modelo físico da rede: O sistema INTI assume que a rede consiste em um conjunto T composto por n objetos (nós) identificados por $\{n_1, n_2, n_3, ..., n_i\}$, onde $n_i \in T$. Cada nó n_i possui um identificador (ID) único. A comunicação ocorre através do meio sem fio utilizando um canal de forma assíncrona, sujeita à perda de pacotes devido à mobilidade dos nós. O raio de transmissão empregado é igual para todos os nós da rede. Assume-se que inicialmente todos os dispositivos começam livres assumindo três papeis: nó membro, um nó associado ou um nó líder. Um nó livre é aquele que não pertence a um agrupamento, este movimenta-se dentro de uma determinada área de cobertura da rede. Um nó membro é aquele que pertence a um agrupamento, ele envia informações para seu líder em um intervalo de tempo. Os nós associados são os nós que fazem a conexão entre agrupamentos para o encaminhamento de dados. Os nós líderes tem a função de coletar as informações dos nós membros e encaminhar para o destino.

Modelo de comunicação: Para que os dispositivos da rede interajam é necessário de protocolos respeitem as limitações dos dispositivos que compõem a IoT. O protocolo 6LoWPAN permite o roteamento de pacotes IPv6 na rede 6LoWPAN (IPv6 com baixo consumo de energia para Rede sem fio de Área Pessoal) de uma forma comprimida. A compressão é necessária para permitir a ligação do 6LoWPAN e protocolo de camada física, IEEE 802.15.4. Com esta rede é possível conectar dispositivos com recursos limitados com a Internet convencional para formar a IoT. O RPL (Protocolo de roteamento IPv6 para redes de baixa potência e com perdas) [Gaddour and Koubâa 2012] é um protocolo de roteamento que respeita as limitações dos dispositivos da IoT. A desvantagem deste protocolo é que ele funciona só em ambientes estáticos [Korbi 2012]. O protocolo de comunicação utilizado pelo sistema INTI é uma variação do protocolo RPL, onde considera-se mobilidade e formação de agrupamentos. Além disso, a conexão orientada protocolos da Web, como HTTP não são viáveis e um novo protocolo, o protocolo de aplicação restrita (COAP), tem sido padronizada para a IoT.

Modelo do ataque na rede: Cada nó é responsável pelo envio e encaminhamento dos pacotes de dados. Um ataque tem como objetivo afetar o funcionamento normal e pôr em perigo a segurança da rede. Os nós afetados pelo ataque desempenham a função de nó líder, nó associado ou nó membro. O ataque *sinkhole* anuncia para seus vizinhos que possui o caminho ideal, o mais curto para o destino pretendido, a fim de atrair a maior quantidade de tráfego de uma certa área prejudicando um ponto de coleta de dados. Além disso, este ataque realiza outros tipos de ameaças, como o ataque *selective forwarding*.

4. Arquitetura do Sistema INTI

Esta seção apresenta a arquitetura INTI (*Sistema de detecção de **I**ntrução de ataques Si**N**khole sobre 6LoWPAN para a Interne**T** das Co**I**sas*) e detalha seus módulos. O INTI considera a mobilidade dos dispositivos (nós) que fazem parte da IoT, bem como a adaptabilidade dos nós adversários, sendo estes totalmente distribuídos e reativos. O INTI possui quatro módulos: o módulo de formação e restauração dos agrupamentos, o módulo de monitoramento, o módulo de detecção, e o módulo isolamento, ilustrado na Figura 2.

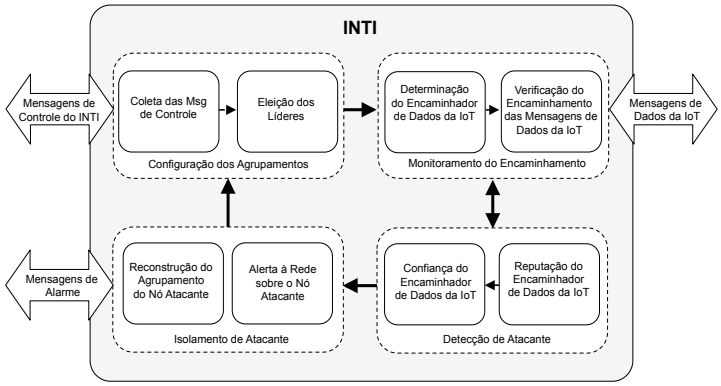


Figura 2. Arquitetura do INTI

O sistema INTI tem as propriedades de *auto-organização* e *auto-reparação*. A propriedade de auto-organização tem como objetivo a coordenação e cooperação dos dispositivos para a configuração da rede. Já a auto-reparação auxilia na detecção de um nó falho, reagrupando os nós afetados, a fim de manter a estabilidade da rede.

4.1. Formação e restauração dos agrupamentos

Este módulo gera uma hierarquia baseada em nós líderes para a formação de agrupamentos a fim de organizar, garantir a escalabilidade e estender a vida útil da rede. Os nós da rede são classificados como: *nós membros*, *nós associados* e *líderes*, conforme Figura 1. A classificação de cada nó mudará dependendo da função que desempenhe dentro da rede.

Inicialmente todos os nós da rede começam livres transmitindo e coletando dados de controle, como ilustrado na Figura 3. O nós enviam dados via *broadcasts* a fim de estabelecer troca de mensagens. Essas mensagem estimam a quantidade de nós vizinhos para eleger os líderes. Os nós livres são classificados como nós líderes quando estes possuem a maior quantidade de nós vizinhos em relação aos outros. Após da eleição dos líderes, são definido os agrupamentos. Nesta fase, os líderes aguardam a decisão dos nós livres (vizinhos), sendo estes nós responsáveis por selecionar um dos líderes para formar o agrupamento. Uma vez formado os agrupamentos, os líderes verificam se um dos nós de seu agrupamento (nós membros) recebeu mais mensagens de diferentes líderes. Caso exista um nó membro que recebeu diferentes mensagens, este nó será classificado como nó associado, sendo este capaz de interligar agrupamentos. No caso de haver dois nós membros dentro da mesma área, o nó membro é escolhido sendo aquele nó que possui a maior índice de energia (*IE*) que é determinado por: $IEi = \frac{TEr_i}{TEc_i}$, onde TEr_i é o total de energia restante do mesmo nó n_i e TEc_i representa o total de energia consumida pelo nó.

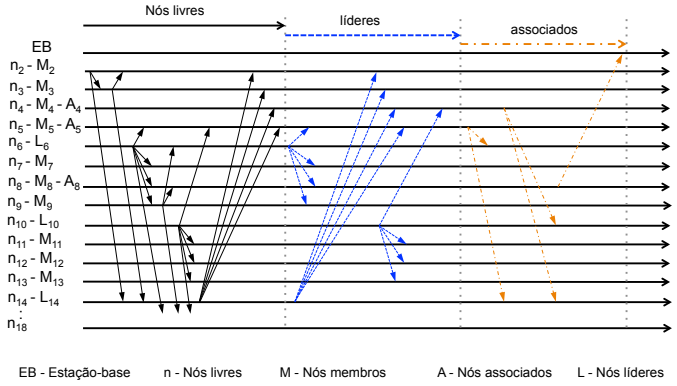


Figura 3. Formação dos agrupamentos no INTI

O uso da função densidade de probabilidade Beta denotada por $Beta(p|\alpha,\beta)$, representa o status de um nó dentro da IoT. Além disso, os parâmetros $Beta(\alpha,\beta)$ são constantemente atualizados determinando o comportamento de um nó. A Equação 1 define a função Beta, em que *p* é a probabilidade de ocorrência de α e $(1-p)$ é a probabilidade de ocorrência de β.

$$Beta(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)} \quad (1)$$

$$Onde: 0 \leq p \leq 1 \quad e \quad \alpha,\beta > 0$$

A probabilidade de densidade e sua expectativa estatística fundamenta-se na função Beta. Ela é representada pela integral definida por: $B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-p)^{\beta-1}dt$. A variável *status* (*St*) armazena o comportamento dos nós determinando como o

nó atua na transmissão de mensagens. $St = \frac{\alpha}{\alpha+\beta}$. Este valor tem como base a probabilidade de esperança futura *E(p)*, que é calculado a partir da função de densidade Beta.

A restauração do agrupamento acontece quando um dos nós falha, abandona o agrupamento ou quando ocorre um ataque *sinkhole*. Se um nó líder é afetado por alguns destes problemas, efetua-se uma nova eleição ou os nós membros afetados reagrupam-se em agrupamentos vizinhos. Se um nó associado é afetado existe a possibilidade de escolher outro nó associado, desde que este esteja dentro da área em comum. Caso contrario, se ambos líderes estão dentro do mesmo raio de transmissão, realiza-se uma fusão dos agrupamentos, considerando a maior quantidade de nós membros que cada agrupamento possui. Este método tem como finalidade minimizar o número de líderes.

4.2. Monitoramento do encaminhamento de dados

O módulo de monitoramento aplica os princípios de *watchdog* que monitora e contabiliza o número de transmissões de entrada e saída realizadas por um nó. Para isso, o nó monitor computa a quantidade de transmissões realizadas por um nó "superior" em relação a suas próprias mensagens. Um nó é chamado de nó superior quando este possui um (*rank*) menor. Feito isto, estima-se a quantidade de transmissões realizadas de entrada e saída. Se a quantidade de transmissões de entrada são iguais ao número de transmissões de saída o nó é considerado bom. Caso contrario, o componente assume que está acontecendo algum desvio do seu funcionamento normal.

4.3. Detecção de ataque Sinkhole

No módulo de detecção, o INTI identifica e revela a identidade do nó atacante *sinkhole*. Para isso, este módulo realiza avaliações da reputação e confiança dos nós para detectar nós atacantes. Tais avaliações ocorrem de forma constante mantendo a segurança e a integridade dos nós da IoT. A reputação é a opinião ou percepção que uma entidade cria através de iterações, ações ou informações. Sendo estas iterações de modo diretas ou indiretas com base a tarefas passadas. O uso da distribuição *Beta (α,β)* é essencial para representação da reputação e da confiança dos dispositivos (nós) da IoT. A vantagem de usar esta distribuição é que os parâmetros são continuamente atualizados.

O sistema INTI calcula três predições: incerteza (*i*), crença (*c*) e descrença (*d*) a partir da distribuição *Beta (α,β)* para representar a reputação. Os líderes, nós associados e algumas vezes pelos nós membros realizam esses cálculos. O cálculo destas predições $(i, c, d) \in (0, 1)^3 : i + c + d = 1$ respetivamente. A incerteza é a variância normalizada da distribuição Beta, sendo calculada de acordo com: $i = \frac{12*\alpha*\beta}{(\alpha+\beta)^2*(\alpha+\beta+1)}$, onde α e β obtidas da distribuição Beta. A certeza é computada por $(1 - i)$, que é dividida na crença (c) e na descrença (d) de acordo com sua proporção de iterações de prova. Sendo estas computadas através do valor esperado da distribuição Beta. Este cálculo é obtido por: $c = \frac{\alpha}{(\alpha+\beta)}(1 - i)$. Por ultimo, a descrença (d) é alcançado por: $d = (1 - i) - c = \frac{\beta}{(\alpha+\beta)}(1 - i)$.

Após obtidos os cálculos das predições (i, c, d) é possível calcular a reputação. A reputação de um nó é calculado a partir das próprias experiências baseadas nas predições computadas e do *status* enviado por um nó membro a seu líder. Desta forma, cada nó propagará seu *status* (*St*) sobre **seu comportamento na transmissão de mensagens** para o cálculo de sua reputação. Esses valores são dados de entrada para o uso da teoria de *Dempster-Shafer*, a fim de aumentar a probabilidade de detecção e reduzir os falsos

alarmes. A reputação é um valor continuo dentro dos limites R[0,1], se o valor de um nó é maior ou igual 0,5 considera-se um nó bom caso contrario, é considerado um nó atacante. Um nó $n_i : \Omega\{T, \overline{T}\}$, onde Ω tem três hipótese (*H*): $H = T$ representa que n_i é bom, $\overline{H} = \overline{T}$ mostra que n_i não é bom e $U = \Omega$ em que n_i representa que é bom ou não bom. Por exemplo, se o nó líder L_1 afirma que nó membro m_2 é bom, então a sua atribuição básica de probabilidade é representada na Equação 2.

$$\begin{aligned} m_2(H) = &\quad c \\ m_2(\overline{H}) = &\quad 0 \\ m_2(U) = &\quad 1 - c \end{aligned} \tag{2}$$

Se o nó líder L_1 afirma que nó membro m_2 não é bom, então a sua atribuição básica de probabilidade é representada na Equação 3.

$$\begin{aligned} m_2(H) = &\quad 0 \\ m_2(\overline{H}) = &\quad c \\ m_2(U) = &\quad 1 - c \end{aligned} \tag{3}$$

As probabilidades prévias determinadas pelo líder para o nó m_2 levam em consideração o (*St*) do próprio nó. A construção das probabilidades do nó líder em relação ao nó m_2, conforme mostra a Equação 4, onde K representa a normalização das crenças, sendo representado por $K = \sum_{L \cap M = \emptyset} m_1(L)m_2(M)$, e onde a reputação é dado pelo valor de $m_1(H) \oplus m_2(H)$, sendo este um valor continuo entre $0 \leq m_2 \leq 1$. Este resultado considera $m_2 < 0,5$ como nó com má reputação e com valor de $0,5 \geq m_2$ representará um nó bom.

$$\begin{aligned} m_1(H) \oplus m_2(H) = &\quad \frac{1}{K}[m_1(H)m_2(H) + m_1(H)m_2(U) + m_1(U)m_2(H)] \\ m_1(\overline{H}) \oplus m_2(\overline{H}) = &\quad \frac{1}{K}[m_1(\overline{H})m_2(\overline{H}) + m_1(\overline{H})m_2(U) + m_1(U)m_2(\overline{H})] \\ m_1(U) \oplus m_2(U) = &\quad \frac{1}{K}[m_1(U)m_2(U)], \end{aligned} \tag{4}$$

$$\begin{aligned} Onde : K = &\quad m_1(H)m_2(H) + m_1(H)m_2(U) + m_1(U)m_2(H) + \\ &\quad m_1(\overline{H})m_2(\overline{H}) + m_1(\overline{H})m_2(U) + m_1(U)m_2(\overline{H}) + \\ &\quad m_1(U)m_2(U) \end{aligned}$$

Após obtida a reputação, calcula-se a confiança (*C*). Este cálculo considera dois valores (γ, δ), sendo realizada pela Equação 5, onde u é computado a partir do número de iterações realizadas entre dois nós n_i e n_j, dada por *m*: $u = 1 - \frac{1}{m}$, onde u possui valores entre $0 \leq u \leq 1$.

$$\gamma = u\gamma + r \quad ; \quad \delta = u\delta + (1 - r) \tag{5}$$

A Equação 6 obtém o valor da confiança que varia entre [0,1] com um valor médio de 0,5. Se o valor obtido é maior que 0,5 então o nó é considerado bom caso contrario, o nó é considerado atacante. A reputação, assim como a confiança precisam ser atualizadas de forma constante, para a detecção do *sinkhole*.

$$C = \mathbf{E}(Beta(\gamma + 1, \delta + 1)) = \frac{\gamma + 1}{\gamma + \delta + 2} \tag{6}$$

4.4. Isolamento do atacante

Uma vez detectado um nó *sinkhole*, o módulo de isolamento faz com que ele seja isolado da rede. Para isso, o nó que detectou o *sinkhole* gera e propaga uma mensagem de alarme em *broadcast* com o ID do nó atacante colocando o ID na *blacklist* da estação-base. Além disso, o nó que detectou o ataque promove o isolamento do atacante enviando uma mensagem de restauração para seus vizinhos. O *rank* é o dado que permitirá aos nós realizem a restaurem do agrupamento. Existem três formas de isolar um *sinkhole*: (i) quando um nó *sinkhole* é um nó membro: este será isolado pelo nó líder; (ii) quando o *sinkhole* assume a função de líder; neste caso os nós membros isolam o nó *sinkhole* ou caso exista um nó associado este isola o *sinkhole*; (iii) quando o nó *sinkhole* assume a função de nó associado, este será isolado pelo líder, com o maior *rank*, quebrando a comunicação com ele. É necessário verificar se existe dentro do agrupamento, que isolou o atacante, algum nó associado com o menor *rank*, a fim de encaminhar as mensagens do agrupamento para o destino. Caso contrario, o líder propagará uma mensagem de restauração para os nós do agrupamento para que se juntem em agrupamento vizinhos.

5. Avaliação do INTI

O sistema INTI foi implementado no simulador Cooja que faz parte do Contiki [Dunkels et al. 2004], sendo este um sistema operacional de código aberto para sistemas embarcados e redes de sensores sem fio. O IDS SVELTE usado para comparação também foi implementado neste simulador. Para avaliar a eficácia e a eficiência dos sistemas INTI e SVELTE, foi considerado um cenário com ataques *sinkhole*.

O cenário é composto por 50 nós, alguns fixos e outros moveis que representa a quantidade média de usuários podem transitar em uma estrada. Esses usuários utilizam equipamentos sem fio, como celulares, PDAs, notebooks, e movimentam-se em uma área delimitada. A área demográfica de IoT utilizada compreende um ambiente realística de caráter urbano como uma estrada [Bellavista et al. 2013], onde existe uma mistura de objetos e dispositivos. Esses usuários podem ser pedestres, pessoas correndo, ciclistas até automóveis que movimentam-se com velocidades entre 0 m/s até 6,94 m/s. A quantidade de nós *sinkhole* igual a 10 e 15 o que representa 20% e 30% dos nós pertencentes a ambos sistemas de detecção. Os nós usam o canal sem fio na comunicação, seguindo o modelo de propagação (*Unit Disk Graph Medium* (UDGM)) e o modelo de movimentação aleatória *RandomWaypoint* em uma região de $100x100m$. O protocolo de roteamento empregado no INTI é uma modificação do protocolo RPL, o raio de alcance dos nós varia de 10 a $40m$, e o tipo de pacote de transporte utilizado pelos nós é o UDP. O tempo de simulação é de 1500s. Os resultados apresentados são a média de 35 simulações e com um intervalo de confiança de 95%. As métricas utilizadas pelo sistema INTI são detalhadas a seguir:

Taxa de detecção do ataque sinkhole (T_{det}) contabiliza os ataques identificados corretamente pelo sistema INTI. O cálculo desta métrica é alcançada seguindo a Equação 7, em que *X* representa o total de iterações dos nós atacantes e os respectivos resultados obtidos pelo INTI, dado na forma de *X = (d, c)*, em que *d* é o valor da detecção realizada pelo sistema e *c* é a autêntica condição do nó $n_i \in R$.

$$T_{det} = \frac{\sum D_i}{|X|} \forall_i \in X \quad \text{onde} \quad D_i = \begin{cases} 1, & \text{se} \quad d_i = c_i, \\ 0, & \text{se} \quad d_i \neq c_i. \end{cases} \tag{7}$$

Taxa de falsos negativos (Tx_{Fn}) indica a quantidade de vezes em que os nós *sinkhole* foram considerados pelo sistema como nós confiáveis. Essa métrica é obtida pela Equação 8, em que *X* contabiliza o número total de iterações realizadas pelo INTI e T_{det} representa a taxa de detecção do *sinkhole*, que foi alcançada seguindo a Equação 7.

$$Tx_{Fn} = |X| - T_{det} \tag{8}$$

Taxa de falsos positivos (Tx_{Fp}) determina a quantidade de vezes que o sistema detectou um ataque *sinkhole* sendo este negativo. A Tx_{Fp} é calculada pela Equação 9, em que *Z* é o conjunto das iterações dos nós normais, na forma *Z = (d, c)*, onde *d* representa o valor da detecção realizada pelo INTI e *c* é a condição real do nó $n_i \in R$, onde *c=1* representa um nó atacante e *c=0* representa um nó bom.

$$Tx_{Fp} = \frac{\sum Dp_i}{|Z|} \forall_i \in Z \quad \text{onde} \quad Dp_i = \begin{cases} 1, & \text{se} \quad d_i = 1, \\ 0, & \text{se} \quad d_i \neq 0. \end{cases} \tag{9}$$

Consumo de energia (E_{gc}) indica o total do consumo de energia dos nós da rede durante a simulação. Este cálculo é representado pela Equação 10, em que $\sum_{z=1}^{i} TE_i$ representa a somatória total de energia inicial de todos os nós da rede e $\sum_{z=1}^{i} TE_r$ é o somatório total da energia restante dos nós. Onde $\sum_{z=1}^{i} n_z = 1$ e $\forall$ *R* obtendo assim a energia total consumida quando é rodado o sistema.

$$E_{gc} = \sum_{z=1}^{i} (TE_i - TE_r) \tag{10}$$

Taxa de entrega de pacotes $(Tx_{Entrega})$ determina o total de pacotes de dados recebidos com sucesso. O cálculo da $Tx_{Entrega}$ é apresentada na Equação 11, onde esta é calculada dividendo o número de pacotes recebidos pelo destino através do número de pacotes originados pela origem.

$$Tx_{Entrega} = \frac{NpacotesRecibidos}{NpacotesEnviados} X100 \tag{11}$$

5.1. Eficácia

A avaliação da eficácia do INTI e SVELTE considera as métricas a T_{det}, Tx_{Fn} e Tx_{Fp}. No cenário fixo, o INTI e o SVELTE apresentam praticamente uma igualdade (92% e 90% respetivamente) na detecção de ataques *sinkhole*, como ilustra a Figura 4(a). Essa diferença de detecção entre o INTI e o SVELTE ocorre porque o SVELTE tem que percorrer todos os nós da rede, a fim de detectar as inconsistências. Em um cenário móvel, como ilustra a e Figura 4(b), a taxa de detecção do SVELTE diminuiu para 24% e a da INTI é superior a 70%. Esse aumento na taxa de detecção entre o INTI e o SVELTE se deve ao fato que o SVELTE não permite a mobilidade dos nós, sendo uns dos seus pontos fracos. Logo, o INTI supera ao SVELTE em um cenário fixo como móvel.

A taxa de falsos negativos obtidos pelo INTI em um cenário fixo é de 8%, Figura 5(a). Isso significa que poucos nós *sinkhole* não são detectados. A falha na detecção de um *sinkhole* pode acontecer devido à autonomia na detecção, que permite que os nós

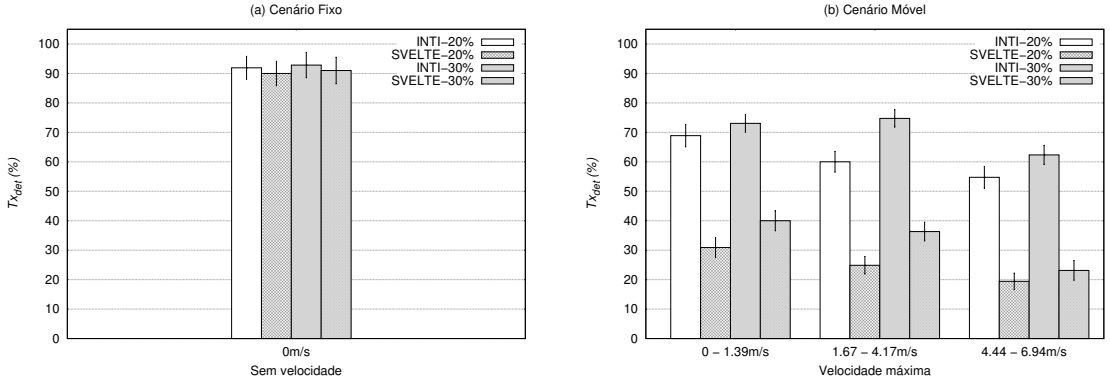


Figura 4. T_{det} do INTI e SVELTE diante de ataques sinkhole

contabilizem individualmente os pacotes transmitidos por outro nó, atuando como observador. Dessa forma, alguns nós podem demorar na identificação de nós *sinkhole*. Para um cenário com nós móveis, a quantidade de falsos negativos obtida pelo INTI é de 28% e pelo SVELTE é de 38%, conforme apresentado na Figura 5(b). Esse aumento de falsos negativos acontece pela dinamicidade dos nós da rede.

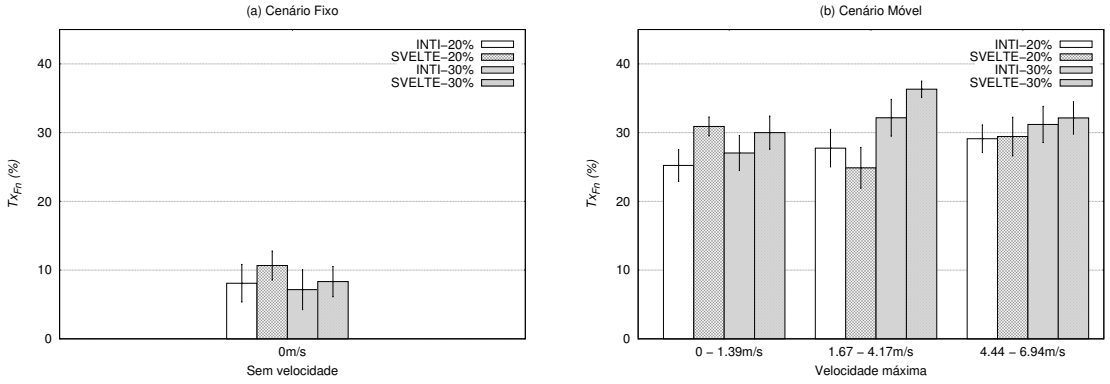


Figura 5. Tx_{Fn} do INTI e SVELTE diante de ataques sinkhole

Como mostra a Figura 6(a), a taxa de falsos positivos obtida pelo INTI na detecção de ataques *sinkhole* em um cenário com nós fixos é inferior a 3%. Enquanto, o SVELTE alcançou uma taxa em média de 4%. No cenário com nós móveis, Figura 6(b), a taxa de falsos positivos obtida pelo INTI é inferior a 30%, sendo que no SVELTE é de aproximadamente 39%. As detecções erradas podem acontecer quando alguns nós que reencaminham os pacotes de outros nós atrasam-se. Assim, momentaneamente eles são considerados *sinkhole*, porém conforme acontece a movimentação e a interação entre os nós, eles são identificados como nos bons.

5.2. Eficiência

As métricas da eficiência verificam o desempenho obtido pelo INTI, essas métricas são: E_{gc}, $Tx_{Entrega}$ e as funções assumidas pelos nós dentro da rede como: número de agrupamentos, número de líderes, número de associados, número de nós por líder e o número de nós solitários, a fim de determinar o desempenho do INTI para se adaptar às variações do ambiente. A métrica empregada para a avaliação do desempenho é o consumo de energia E_{gc}, como mostram os gráficos na Figura 7. No cenário fixo, o INTI apresenta um consumo de energia de 25000(mj), sendo menor ao consumo do SVELTE de 67000(mj),

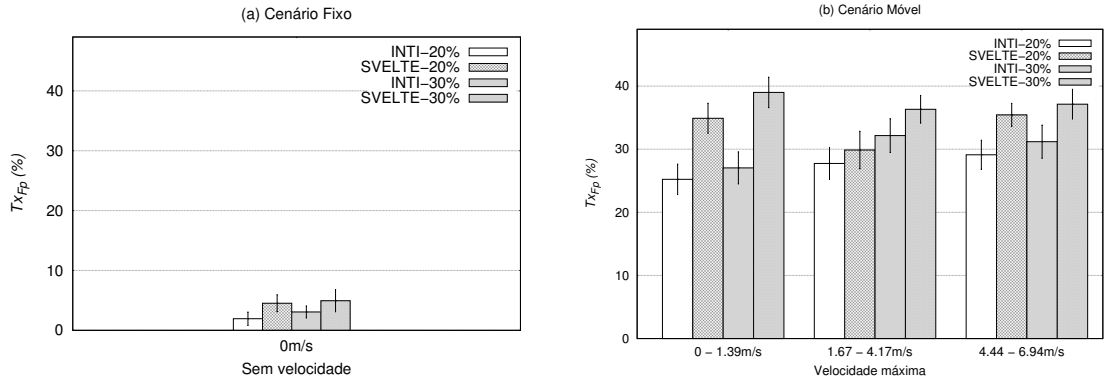


Figura 6. Tx_{Fp} do INTI e SVELTE diante de ataques sinkhole

como ilustrada na Figura 7(a). Em um cenário móvel, o INTI obtém quase o mesmo consumo que em um cenário fixo. Isto se deve à técnica usada pelo INTI permitindo a formação de agrupamentos para diminuir o consumo de energia e a escolha do nó associado como o maior índice de energia (*IE*). É interessante observar que o consumo de energia do SVELTE em um cenário móvel aumento para 75000(mj). Este aumento é devido à formação da topologia da rede no SVELTE, conforme o mostrado na Figura 7(b).

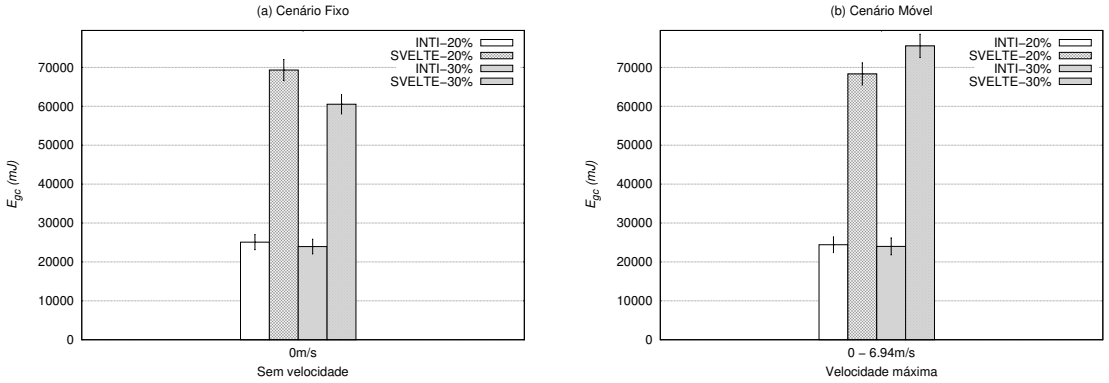


Figura 7. E_{gc} do INTI e SVELTE diante de ataques sinkhole

No cenário fixo, o SVELTE apresenta uma maior taxa de entrega $Tx_{Entrega}$ alcançando o 99% na entrega de dados da IoT, superando aos 95% alcançado pelo sistema INTI, como ilustrado no gráfico da Figura 8 (a). É possível também observar que o sistema INTI começa com uma taxa de entrega de 79% conseguindo aumentar 95%, essa variação é devido à pouca quantidade de nós dentro da área estabelecida. Desta forma, com o aumento da quantidade de nós a taxa de entrega aumenta. O gráfico da Figura 8 (b) apresenta só a avaliação do sistema INTI, já que o sistema SVELTE não permite a mobilidade dos nós. Este gráfico considera diferentes velocidades definidos anteriormente. Como pode-se apreciar o INTI no começo possui uma taxa de entrega superior a 55% mais conforme aumenta a quantidade de nós e a velocidade o INTI aumenta conseguindo alcançar uma taxa de entrega superior a 75%.

Outra métrica considerada é: **o número de agrupamentos, o número de líderes, o número de associados, o número de nós por líder e o número de nós solitários** A Figura 9 (a) ilustra a quantidade de agrupamentos, número de líderes, número de associados, número de nós por líder e o número de nós solitários calculados em um cenário fixo

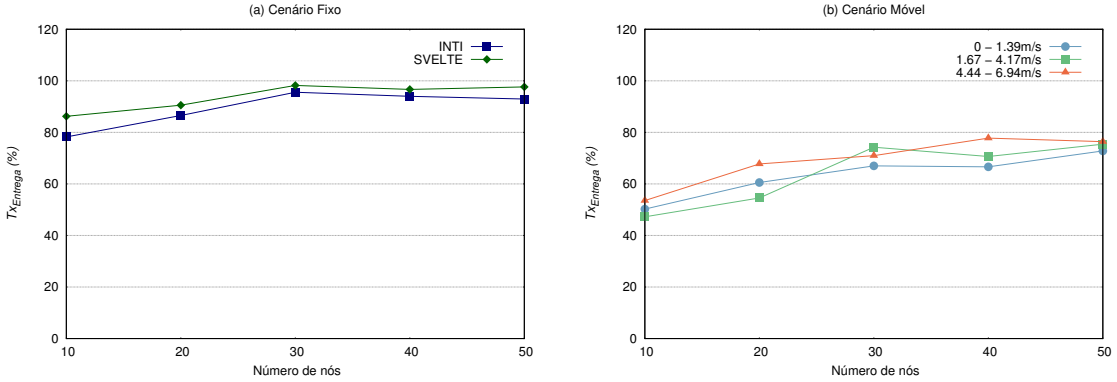


Figura 8. $Tx_{Entrega}$ do INTI e SVELTE

durante a simulação. No cenário móvel, o INTI apresenta uma redução na quantidade de nós que desempenham alguma função no encaminhamento de dados. Além disso, o número de nós solitários aumenta, como mostrado no gráfico da Figura 9 (b), isto é a causada da mobilidade dos nós. Sendo que estes nós movimentam-se dentro de uma área determinada, entrando e saindo dos agrupamentos formados.

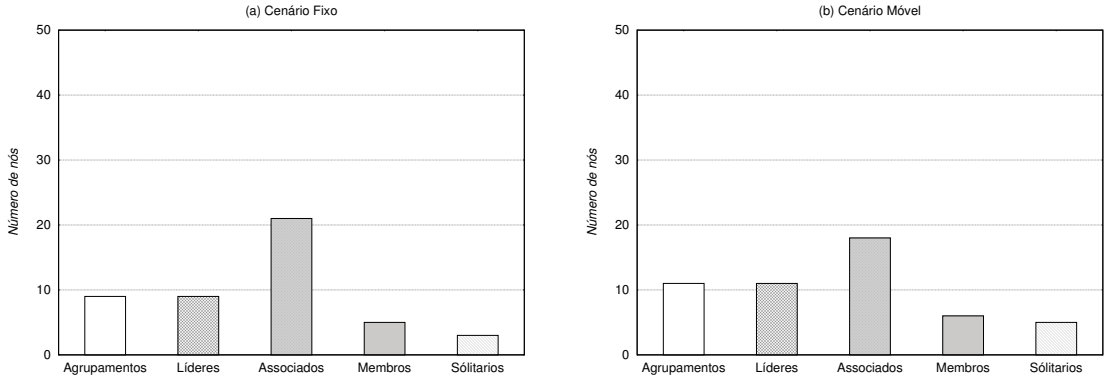


Figura 9. Funções assumidas pelos nós na IoT

6. Conclusão

Este artigo propôs o sistema INTI para a detecção e isolamento de ataques *sinkhole* na IoT. O INTI usa uma abordagem baseada no comportamento dos nós durante a transmissão das mensagens. Este comportamento é definido pela reputação e confiança de cada nó. O INTI foi avaliado em um cenário realístico para o uso da IoT, e os resultados obtidos mostram que ele alcançou uma taxa de detecção de ataques *sinkhole* de até 92% em um cenário com nós fixos e de 75% em um cenário com nós móveis. Além disso, o INTI apresentou um baixo consumo de energia e uma baixa taxa de falsos positivos e negativos em relação ao SVELTE. Como trabalhos futuros, avaliaremos a eficácia do INTI na detecção de outros tipos de ataques que acontecem na IoT.

Referências

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. volume 54, páginas 2787–2805, Catania, Itália. Elservier Science Publishers B. V.

Bannack, A., da Silva, E., Lima, M. N., dos Santos, A. L., and Albini, L. C. P. (2008). Segurança em redes ad hoc. *Anais do XXVI Simpósio Brasileiro de Telecomunicaçoes (SBRT)*, páginas 19–20.

Bellavista, P., Cardone, G., Corradi, A., and Foschini, L. (2013). Convergence of MANET and WSN in IoT urban scenarios. volume 13, páginas 3558–3567. IEEE.

Dunkels, A., Gronvall, B., and Voigt, T. (2004). Contiki-a lightweight and flexible operating system for tiny networked sensors. páginas 455–462. IEEE.

Gaddour, O. and Koubâa, A. (2012). RPL in a nutshell: A survey. páginas 3163–3178. Elsevier.

Ganeriwal, S., Balzano, L. K., and Srivastava, M. B. (2008). Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):15.

Jin Qi, Tang Hong, K. X. L. Q. (2012). Detection and defence of sinkhole attack in wireless sensor network. In *ICCT-2012*, páginas 809–813, Chengdu, China. IEEE Security.

Kasinathan, P., Pastrone, C., Spirito, M. A., and Vinkovits, M. (2013). Denial-of-service detection in 6lowpan based internet of things. In *WiMob-2013*, páginas 600–607. IEEE.

Keally, M., Zhou, G., and Xing, G. (2010). Watchdog: Confident event detection in heterogeneous sensor networks. In *RTAS-2010 16th IEEE*, páginas 279–288. IEEE.

Korbi, I.E. Ben Brahim, M. A. C. S. L. (2012). Mobility enhanced RPL for wireless sensor networks. In *NOF-2012*, páginas 21–23. IEEE.

Lima, M., dos Santos, A. L., and Pujolle, G. (2009). A survey of survivability in mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 11(1):66–77.

Moon, S. Y. and Cho., T. H. (2009). Intrusion detection scheme against sinkhole attacks in directed diffusion based sensor networks. In *IJCSNS*, páginas 118–122, Coreia. IEEE Computer Society.

Perez-Toro, C. R., Panta, R. K., and Bagchi, S. (2010). Rdas: reputation-based resilient data aggregation in sensor network. In *IEEE SECON-2010*, páginas 1–9. IEEE.

Raza, S., Wallgren, L., and Voigt., T. (2013). Svelte: Real-time intrusion detection in the internet of things. páginas 2661 – 2674, USA. Elservier.

Sedjelmaci, H. and Feham, M. (2011). Novel hybrid intrusion detection system for clustered wireless sensor network. In *International Journal of Network Security & Its Applications*.

Shafiei, H., Khonsari, A., Derakhshi, H., and Mousavi, P. (2014). Detection and mitigation of sinkhole attacks in wireless sensor networks. volume 80, páginas 644–653. Elservier.

Sheela, D., K. C. N. and Mahadevan., G. (2011). A non cryptographic method of sinkhole attack detection in wireless sensor networks. In *ICRTIT-2011*, páginas 527–532, Tamil Nadu, Chennai. IEEE Security.

Wahab, O. A., Otrok, H., and Mourad, A. (2014). A cooperative watchdog model based on dempster–shafer for detecting misbehaving vehicles. *Computer Communications*, 41:43–54.

Wallgren, L., Raza, S., and Voigt, T. (2013). Routing attacks and countermeasures in the rpl-based internet of things. *International Journal of Distributed Sensor Networks*, 2013.

Identificação e Caracterização de Comportamentos Suspeitos Através da Análise do Tráfego DNS

Kaio R. S. Barbosa[1], **Eduardo Souto**[1], **Eduardo Feitosa**[1], **Gilbert B. Martins**[1]

[1]Instituto de Computação - Universidade Federal do Amazonas (UFAM)
Av. Rodrigo Otávio Jordão Ramos, 3000, Coroado.
CEP 69077-000, Manaus-AM, Brasil

{kaiorafael,esouto,efeitosa,gilbert.martins}@icomp.ufam.edu.br

Abstract. *The Domain Name System (DNS) provides mechanisms for translating domain names into IP address. This service is used by both legitimate users and suspicious applications which may request mail servers' address before sending spam. This paper presents a methodology based on graph theory that distinguishes between legitimate and malicious traffic queries patterns. Name resolutions are modeled in a graph that illustrates the communication patterns between hosts and how the queries were held. To validate the proposal, the .br DNS domain traffic is investigated. The results show a 35% reduction of the hosts to be analyzed and the presence of suspicious behavior.*

Resumo. *O Sistema de Nomes de Domínios (DNS) fornece mecanismos para traduzir os nomes de domínios em endereços IP. Este serviço é usado tanto por usuários legítimos quanto aplicações suspeitas que podem solicitar endereços dos servidores de email antes de enviar spam. Este trabalho apresenta uma metodologia que utiliza teoria dos grafos para distinguir padrões entre consultas legítimas e maliciosas no tráfego. As resoluções de nomes são modeladas em um grafo que ilustra o padrão de comunicação entre hosts e como as consultas foram realizadas. Para validação da proposta, o tráfego DNS do domínio .br é investigado. Os resultados mostram uma redução de 35% do total de hosts a serem analisados e a presença de comportamentos suspeitos.*

1. Introdução

O Sistema de Nomes de Domínios (DNS) [Mockapetris 1987] traduz nomes de domínios em endereços IP permitindo que aplicações e usuários tenham acesso aos diversos serviços de rede como sistemas de comércio eletrônico, navegação de sites e envio de emails. Entretanto, devido ao acesso livre e distribuído do protocolo DNS, aplicações maliciosas também podem fazer consultas de nomes de domínios para realizar ataques, tais como negação de serviço, propagação de spam em massa e distribuição de *malwares*. Em todos os casos, o tráfego DNS é consultado inicialmente para obter informações sobre o serviço desejado.

A identificação e distinção de comportamentos benignos e maliciosos no tráfego DNS ainda é um problema em aberto, especialmente quando o tráfego é analisado em servidores de Domínio de Primeiro Nível, também conhecidos como *Top-Level Domains - TLD* e servidores Raiz (*Root Servers*) [Castro et al. 2008]. As possíveis razões são devidas ao volume de tráfego a ser processado ou a questões legais de acesso

[Antonakakis et al. 2010]. Além disso, a distinção de comportamentos é agravada quando aplicações maliciosas mudam de comportamento para dificultar a sua detecção. Um típico exemplo de tal situação é o *worm Conficker* que acessa uma lista diária de novos domínios gerados dinamicamente por algoritmo [Shin e Gu 2010].

Embora algumas aplicações maliciosas consigam evadir os sistemas de detecção, ainda é possível detectar comportamento anômalo no tráfego DNS devido ao ciclo de vida das máquinas infectadas (*bots*), pois em algum momento, os *bots* utilizam o tráfego DNS para coletar informações antes do ataque. Desta forma, é possível assumir que hosts maliciosos apresentem padrões de comportamentos similares na rede. Por exemplo, antes de enviar um email, é necessário que o atacante realize uma consulta DNS para obter o endereço do servidor que recebe esse tipo de mensagem. Um abuso nesse tipo de consulta pode denotar o comportamento de *SpamBot* [Ishibashi et al. 2005]. Da mesma forma, um ataque de reconhecimento de rede, o tráfego DNS é consultado para identificar hosts que são válidos e ativos [Barbosa e Souto 2009].

Motivado pelos problemas citados acima, este trabalho apresenta uma metodologia que utiliza Teoria dos Grafos para identificar padrões de comportamentos suspeitos em tráfego TLD. Elementos da consulta DNS como IP de origem, nome de domínio e registro de recurso (RR - *resource record*) são modelados em um grafo direcionado que corresponde à consulta partindo do endereço IP de origem para o nome de domínio, e o registro de recurso denota o objetivo da consulta realizada. A metodologia proposta permite aos operadores de rede entender a relação de comunicação entre hosts e facilmente indicar aqueles hosts que possuem maior relevância para investigação. Tal abordagem minimiza não só a quantidade de hosts a ser analisada bem como o tempo necessário para a análise.

Para avaliar a metodologia proposta, este trabalho investiga as consultas recebidas por servidores de primeiro nível (TLDs) do domínio `.br`, disponíveis no projeto DITL (*Day in The Life of the Internet*)[DITL 2014]. Consultas DNS em TLDs fornecem uma visão em larga escala de como o tráfego DNS de uma região é utilizado. Por exemplo, caso um host envie spam em massa para diferentes regiões geográficas no globo, os destinatários deste email solicitarão do servidor de domínio do país de origem (TLD) mais informações sobre o remetente. Desta forma, ao observar consultas DNS em servidores de domínios de primeiro nível, é possível entender como um domínio é requisitado por hosts na Internet e por hosts dentro da zona de domínio.

Diferente de outros trabalhos, a metodologia proposta aqui utiliza apenas as solicitações DNS como fonte de análise. Além disso, os resultados mostram que a metodologia proposta reduz em média 35% dos hosts a serem analisados por operadores de redes. Devido à atividades supostamente suspeitas, clientes de banda larga - usuários domésticos - são os principais hosts consultados por servidores de email e servidores recursivos DNS (rDNS). Uma validação mostrou que 93% dos hosts identificados como relevantes estavam cadastrados em listas negras. Da mesma forma, é possível identificar um conjunto de características que podem ser utilizadas para geração de filtros e classificação automática de tráfego anômalo.

O restante desse trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados à identificação de comportamentos suspeitos através do

tráfego DNS; a Seção 3 demonstra como a Teoria dos Grafos é utilizada no processo de detecção de anomalias; a Seção 4 apresenta os resultados obtidos, descrevendo o padrão de comunicações e evidências de *bots* para o envio de spam; a Seção 5 apresenta as conclusões sobre os resultados e apresenta novos direcionamentos e trabalhos futuros.

2. Trabalhos Relacionados

A análise passiva do tráfego DNS de servidores raiz (*root servers*), servidores de Domínios de Primeiro Nível (chamados de *Top-Level Domains*, ou TLDs) como `.com`, `.net` e `.gov`, e de servidores responsáveis pelos domínios de código de país (*country code TLDs*, ou ccTLDs) permite entender como a resolução de nomes é utilizada globalmente.

Muitos trabalhos têm como objetivo entender e identificar as principais características do tráfego de domínio de primeiro nível, incluindo frequência de registros de recurso, consultas mal formadas e domínios inválidos. Por exemplo, para entender como o tráfego DNS é utilizado por servidores raiz, Castro et al. (2008) analisam o tráfego DNS de servidores raiz coletado em Janeiro de 2006, Janeiro de 2007 e Março 2008 através do projeto DITL (*Day in the Life of the Internet*)[DITL 2014]. Os resultados mostram como os registros de recursos são distribuídos entre as consultas e que existe uma grande quantidade de consultas inválidas que não deveriam alcançar esses servidores, como consultas com TLDs inválidos, consultas recursivas e consultas onde o nome do domínio é um endereço IP (`A-to-A`).

Outros trabalhos direcionam essa análise para servidores de domínios de código de país. Em 2009, Barbosa e Souto (2009) analisaram o tráfego do domínio `.br` coletado em dois dias do mês de março de 2008. Entre os resultados, os autores observaram que as consultas do tipo `PTR` , utilizadas para informar o nome de um domínio a partir de um endereço IP, representam 42,91% do total de consultas. Por outro lado, a fração de consultas do tipo `A` , que corresponde aos registros que mapeiam nomes de máquinas para endereços IP dos hospedeiros, representa 30,29% do total de tráfego analisado. Devido ao volume de tráfego `PTR` em relação ao tipo `A` , os autores sugeriram que tal comportamento era resultado de atividades suspeitas no tráfego, pois outros trabalhos que investigam o tráfego TLD apontam que o registro `A` é o mais frequente [Castro et al. 2008, Yuchi et al. 2009]. Por esse motivo, este trabalho segue com uma análise mais detalhada das consultas destinadas ao domínio `.br` observando agora tráfegos coletados no projeto DITL em 2008, 2009 e 2010 [1].

Antonakakis et al. (2010) apresentam um sistema de reputação dinâmica para nomes de domínios novos ou desconhecidos no ccTLD do Canadá (`.ca`). O sistema, denominado `Notos`, analisa o comportamento do tráfego DNS e atribui uma pontuação de acordo com as atividades relacionadas com o domínio investigado. O comportamento histórico do DNS é obtido a partir de bases legítimas e maliciosas. O comportamento legítimo é coletado em servidores recursivos DNS, enquanto o tráfego malicioso é obtido através de sensores de rede como *honeypots*, *spam-traps* e *sandboxes*. Para facilitar a análise do tráfego, os autores agrupam domínios de comportamentos semelhantes observando características léxicas do nome de domínio (e.g.: frequência de caracteres e tamanho do nome) e de rede (e.g.: número AS e total de endereços IPs associados ao

[1] O Brasil participou do projeto DITL até o ano de 2010.

domínio). A partir dessas características, nomes de domínios que apresentem os comportamentos conhecidos podem ser classificados como benignos ou suspeitos. O trabalho de Choi e Lee (2012) também utiliza o tráfego DNS para agrupar comportamentos semelhantes. Os autores observam consultas repentinas, quantidade de consultas únicas, domínio requisitado por múltiplos IPs de origem em um determinado intervalo de tempo e tentam identificar nesses padrões de consultas redes maliciosas (*botnets*).

Ramachandran et al. (2006) utilizaram análise de grafo para identificar consultas às listas negras originadas por membros de *botnets*. Os autores observaram a relação entre o número de consultas enviadas e recebidas por essas listas. A partir dessa relação, os autores determinaram limiares que permitiram identificar comportamento suspeito no tráfego DNS. Ramachandran et al. validaram a proposta utilizando uma base de *spamtrap*, a qual possuía endereços de *bots* conhecidos. Tais endereços foram observados consultando listas negras para identificar se outros *bots* da mesma rede, ou o próprio endereço, estavam cadastrados nessas bases de dados.

De maneira geral, este trabalho tem como objetivo entender o padrão de comunicação entre os hosts na rede através do protocolo DNS e como esse tráfego é utilizado. Para entender esse padrão, os registros de recursos são utilizados como filtro para identificar e extrair comportamentos de hosts relevantes para análise. Por exemplo, o registro `MX`, que indica uma lista de servidores que devem receber emails de um domínio, é frequentemente utilizado para encontrar servidores de email abertos e suscetíveis aos ataques de spam em massa [Ishibashi et al. 2005]. Em alguns casos, o tipo `A` também pode ser utilizado para detectar domínios de spam [Jiang et al. 2010]. O registro reverso (`PTR`) é útil para identificar endereços IPs válidos, pois cada IP acessível na Internet deve possuir um nome reverso [Barr 1996]. Em ataques de reconhecimento de rede, o registro `PTR` é mais utilizado [Kumagai et al. 2010]. Ataques de dicionário contra o serviço `SSH` podem ser identificados através da distribuição de frequência do registro reverso [Shibata et al. 2012]. Por esses motivos, os tipos de registros de recursos são essenciais para detectar anomalias que utilizam o tráfego DNS.

3. Metodologia

A metodologia proposta consiste em um modelo de representação do tráfego de rede baseada em grafos direcionados que modelam os relacionamentos entre hosts e domínios consultados. As etapas do processo de identificação de consultas maliciosas a partir da análise do tráfego DNS são:

1. Construção do grafo original, onde o tráfego DNS é modelado em grafo direcionado. Os vértices são formados por hosts e nomes de domínios e as arestas, as comunicações entre os vértices.
2. Transformação do grafo, tem objetivo reforçar as conexões do grafo para encontrar padrões de comunicações que não foram modelados inicialmente.
3. Redução do grafo, onde os componentes conexos irrelevantes do grafo transformado são eliminados.
4. Classificação das consultas, onde um conjunto de métricas, definidas para descrever as propriedades estruturais do grafo, é usado para classificar os nós e identificar possíveis comportamentos maliciosos nos hosts associados.

3.1. Construção do Grafo Original

As consultas DNS são modeladas através de um grafo $G = (V, E)$ direcionado, onde V e E são os conjuntos de vértices e arestas de G, respectivamente. Seja A o conjunto de endereços IP de origem e D o conjunto de nomes de domínio. Então, seja $V = A \cup D$, tal que $a \in A$, representa o endereço IP origem da consulta; e $d \in D$ representa o nome do domínio da consulta. Uma aresta $e \in E$, onde $e = (a, d)$, denota a consulta partindo do endereço IP de origem a para o domínio da consulta d. Uma aresta possui o atributo t, tal que $(a, d).t$ denota o tipo de registro de recurso associado à consulta de a para d. Seja $v \in V$, tal que $deg^+(v)$ denota o grau de saída de um vértice e $deg^-(v)$ denota o grau de entrada. O grau do nó $deg(v)$ é denotado pela soma de todos os graus de entrada e saída do vértice. Finalmente, a função $f_{ip}(v)$ retorna o endereço IP de $v \in V$.

Para ilustrar como as consultas DNS são modeladas em um grafo direcionado, considere o exemplo na Figura 1. Por questões de privacidade, endereços privados serão utilizados. A Figura 1 apresenta um conjunto de *Consultas DNS* e o grafo resultante da modelagem, denominado como *Grafo Original*. Por simplicidade, os endereços IPs `192.168.0.1`, `192.168.1.1` e `192.168.3.1` são denotados como `A1`, `A2` e `A3`, e os domínios da consulta `google.com.br`, `example.com`, `1.3.168.192.in-addr.arpa` e `sbc.org.br` são denotados como `D1`, `D2`, `D3`, `D4`, respectivamente. Na primeira linha das consultas DNS, é possível observar o endereço IP `A1` utilizando o registro do tipo `MX` para encontrar o endereço do servidor de email que responde pelo domínio `google.com.br`. Tal consulta é ilustrada pela aresta $(A1, D1)$ no *Grafo Original*. A segunda consulta realizada por `A1` é demonstrada na terceira linha. Tal exemplo denota `A1` buscando pelo endereço reverso (`PTR`) de `1.3.168.192.in-addr.arpa`. A representação dessa consulta é demonstrada pela aresta $(A1, D3)$.

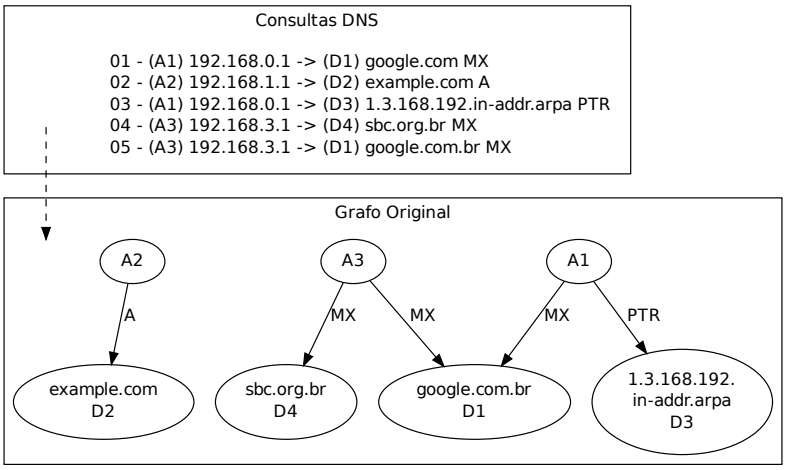


Figura 1. Exemplos de consultas DNS modeladas em Grafo.

3.2. Transformação do Grafo Original

Após a modelagem inicial do grafo, é necessário identificar comportamentos que não foram modelados no *Grafo Original*. Considere o exemplo da consulta na terceira linha em *Consultas DNS*. É possível observar que o nome de domínio (`D3`) foi consultado a partir do registro `PTR` , por isso para identificar se $D3$ também é um endereço IP de origem, isto é, se o host também enviou solicitações, o nome de domínio é convertido do

formato reverso `d.c.b.a.in-addr.arpa` para o endereço IP original `a.b.c.d`. A transformação do grafo tem como objetivo reforçar as conexões dos vértices para representar o padrão de comunicação entre hosts mais próximo da realidade.

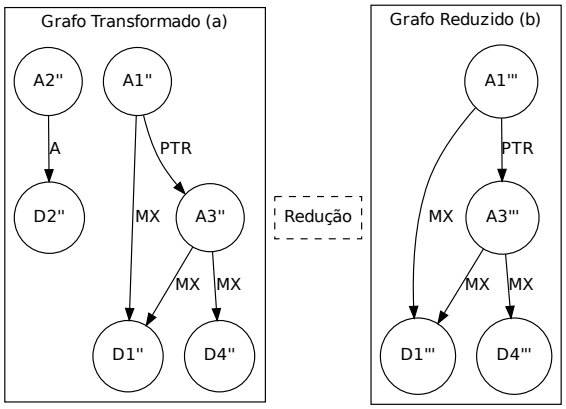


Figura 2. Grafo transformado (a) da Figura 1 e grafo reduzido (b).

Considerando o exemplo da Figura 1, o domínio `D3` equivale ao endereço IP `A3` no formato reverso. Portanto, o grau de entrada de $A3$ no *Grafo Original* resulta em $deg^{-}(A3) = 0$, e o grau de saída $deg^{+}(A3) = 2$. Por outro lado, o grau de $A3$ no *Grafo Transformado* da Figura 2a demonstra o resultado $deg^{-}(A3) = 1$ e $deg^{+}(A3) = 2$. Formalmente, o processo de transformação do grafo é uma função definida pelas seguintes propriedades:

Definição 1: Seja $G = (V, E)$ o *Grafo Original*. O conjunto R é uma coleção de pares ordenados (a_i, a_k), com a_i e $a_k \in V$, gerado pela função $f_1 : V \rightarrow R$ tal que, para todo par ordenado $(a_i, a_k) \in R$ existe um par $(a_i, d_j) \in E$ que foi derivado de uma consulta à um nome de domínio d_j do tipo `PTR`, onde $f_{ip}(a_k) = f_{ip}(d_j)$.

Definição 2: Seja $G = (V, E)$ o *Grafo Original*. Seja R o conjunto gerado de acordo com a Definição 1. O conjunto S é gerado pela função $f_2 : (D, R) \rightarrow S$ tal que, $s \in V$ e para todo $s \in S$ existe um par $(a_i, a_k) \in R$ onde $f_{ip}(s) = f_{ip}(a_k)$.

Definição 3: Seja $G = (V, E)$ o *Grafo Original* e $G' = (V', E')$. Seja R o conjunto gerado de acordo com a Definição 1. Seja S o conjunto gerado de acordo com a Definição 2. Então $G' = (V', E')$ é definido pela função geradora $f_3 : (G, R, S) \rightarrow G'$, tal que $V' = V - S$, $E' \subset (E \cup R)$ e, para todo $(a', d') \in E'$ e todo o $s \in S$, não existe $f_{ip}(d') = f_{ip}(s)$.

3.3. Redução do Grafo Transformado

Finalmente, após o processo de transformação do grafo, é necessário excluir consultas que não são representativas para análise, isto é, solicitações onde o endereço IP de origem comunica-se com o nome de domínio uma única vez, conforme aresta (A2, D2) do grafo transformado (Figura 2a). A Figura 2b ilustra o resultado da redução do grafo. Formalmente, se a_k e d_k constituem um componente conexo de G', então tais vértices são eliminados quando $deg(a_k) = 1$ e $deg(d_k) = 1$. Assim:

Definição 4: Seja $G' = (V', E')$ um grafo gerado de acordo com a Definição 3. O *Grafo Transformado* $G'' = (V'', E'')$ é gerado pela função $f_4 : G' \rightarrow G''$, onde $V'' = V'$

e $E'' \subset E'$ tal que, para todo o par ordenado $(a'', d'') \in E''$, tem-se $deg(a'') > 1$ ou $deg(d'') > 1$.

Definição 5: Seja $G'' = (V'', E'')$ um grafo gerado de acordo com a Definição 4. O *Grafo Reduzido* $G''' = (V''', E''')$ é gerado pela função $f_5 : G'' \rightarrow G'''$, onde $E''' = E''$ e $V''' \subset V''$ tal que, para todo $v''' \in V'''$, tem-se $deg(v''') > 0$.

3.4. Classificação dos Comportamentos Relevantes

O entendimento e classificação de comportamentos através do tráfego DNS são obtidos pela relação entre hosts e como as consultas DNS foram realizadas. Neste trabalho a relação entre hosts é definida pela aresta de pares ordenados e os registros de recursos mostram como os hosts se comunicam. Formalmente, relações de hosts são consideradas relevantes para análise quando o host possui: i) alto grau de entrada do nó - são observados principalmente nós que possuem alta incidência de consultas dos registros de recurso do tipo `PTR` e `MX`, com exceção dos nós com alta incidência natural do tipo `A` como consultas ao `google.com.br` e `facebook.com.br`, pois não representam necessariamente comportamentos suspeitos; ii) alto grau de saída de nó - são observados nós que realizam grandes quantidades de consultas empregando principalmente os registros `PTR`, `MX` e `NS`[2]. Tais tipos de registros de recursos são comumente usados por hosts infectados (*bots*) em diversas atividades maliciosas como envio de mensagens de spam e ataques de reconhecimento. A relação entre hosts também é observada pela iii) razão entre o grau saída e entrada do vértice - a partir dessa relação é possível identificar padrões de consultas semelhantes no tráfego. Por exemplo, servidores de email legítimos são observados no tráfego enviando e recebendo consultas, no entanto, hosts acessando a Internet, geralmente, apenas enviam consultas [Ramachandran et al. 2006]

O entendimento de como as resoluções foram realizadas é obtida pela iv) frequência relativa dos registros de consultas enviadas; e a v) frequência relativa dos registros de consultas recebidas. Tais frequências permitem agrupar hosts que abusam dos registros (e.g: `PTR` e `MX`) independente da quantidade de consultas enviadas. Portanto, a partir desse conjunto de métricas é possível estabelecer uma correlação para definir classes de consultas.

3.5. Classes de Consultas

As Classes de Consultas representam a agregação de hosts relevantes identificados a partir das métricas aplicadas aos vértices. Formalmente, hosts são agrupados pela distribuição estatística de frequência das características individuais. Tal abordagem permite que diferentes comportamentos sejam correlacionados para entendimento do padrão de consultas, ou seja, é possível observar o host em momentos distintos no tráfego DNS. Por exemplo, a razão entre o número de consultas enviadas pelo número de consultas recebidas contribui na análise do tráfego quando a mesma é correlacionada com a frequência de registros de recurso enviados ou recebidos. Por esses motivos, os registros de recursos são fundamentais no entendimento do tráfego.

As classes de consultas são correlacionadas para demonstrar como um host utiliza o tráfego DNS para comunicação. A Tabela 1 apresenta a relação entre as métricas de grafo. Para cada intervalo da classe de consultas a ser investigado, uma métrica é definida

[2]O registro `NS` é utilizado para encontrar o servidor de nomes que responde pela zona de domínio.

Tabela 1. Relação entre as métricas de grafo.

Classe Primária Q	**X**	**Y**	**Z**	**W**
Grau de Entrada (deg^-)	deg^+	Razão	Freq. RR Enviados	Freq. RR Recebidos
Grau de Saída (deg^+)	Razão	deg^-	Freq. RR Enviados	Freq. RR Recebidos
Razão Saída / Entrada	deg^-	deg^+	Freq. RR Enviados	Freq. RR Recebidos
Frequência Registros Recursos Enviados	deg^-	deg^+	Razão	Freq. RR Recebidos
Frequência Registros Recursos Recebidos	deg^-	deg^+	Razão	Freq. RR Enviados

como primária e comparada com as demais métricas do grafo. Por exemplo, seja $Q(deg^-)$ a classe primária, então um vértice é analisado com base no grau de saída (X), razão (Y) e frequência relativa dos registros de recursos enviados (Z) e recebidos (W). Formalmente, um comportamento de um vértice é definido por $Q_{(I(v'''))} = [X, Y, Z, W]$.

4. Resultados

Nesta Seção são demonstrados os resultados obtidos durante experimentos. Primeiramente, a base de dados utilizada na metodologia é descrita. Em seguida, o padrão de comunicação entre os hosts é demonstrado a partir do grau de saída e grau de entrada dos vértices. A classe primária do grau de entrada é utilizada para ilustrar como os hosts relevantes são observados no tráfego. Finalmente, os hosts relevantes são passivamente analisados para demonstrar a validade da metodologia proposta.

Para validar a metodologia proposta foi utilizado tráfego DNS real coletado durante o projeto DITL [DITL 2014], uma cortesia da DNS-OARC (Centro de Pesquisa, Operações e Análise DNS). O tráfego obtido é composto por 15 servidores DNS autoritativos que respondem pelo domínio `.br` , sendo cinco em 2008, quatro em 2009 e seis em 2010. Todos os servidores iniciam a coleta em 00:00:00 UTC e terminam em 23:59:59:9999. A Tabela 2 sumariza os valores na base de dados: dia de coleta; servidores de nome, total de pacotes (consultas e respostas) e o tamanho da base no formato compactado `gzip`.

Tabela 2. Base de dados DITL.

	DITL 2008	DITL 2009	DITL 2010
Dias de Coleta	[18-19]/03	[30-31]/03, 01/04	[14-15]/04
Servidores DNS	{a-e}.dns.br	{a,b,e,f}.dns.br	{a-f}.dns.br
Total de Pacotes	5.3 bilhões	6.4 bilhões	6.9 bilhões
Tamanho da base	229GB	236GB	282GB

Por questões de segurança e privacidade, a análise do tráfego DNS do `.br` deve ser realizada exclusivamente dentro da infraestrutura fornecida pela DNS-OARC. Além disso, tal infraestrutura é compartilhada entre outros pesquisadores que também investigam o tráfego DNS disponível. Por esse motivo, este trabalho investiga o tráfego DNS do servidor `a.dns.br` durante o período 00:00 até 00:59 dos dias 18/03, 30/03 e 14/04 de 2008, 2009 e 2010, respectivamente.

4.1. Características dos Hosts Relevantes

A Tabela 3 apresenta os valores observados durante aplicação da metodologia proposta. O *Total de Consultas* representa todas as consultas observadas durante T, tal que $T = 1h$.

O *Total de Hosts Únicos* demonstra o número de hosts na base antes da identificação do *Total de Hosts Relevantes*. Por exemplo, em 2008 o total de hosts relevantes é 47% menor que o total de hosts únicos. Em 2009 e 2010 essa redução representa 49% e 10%, respectivamente. Isto é, através da metodologia proposta foi possível reduzir em média 35% do total de hosts a serem analisados. Finalmente, a distribuição de frequência dos registros de recurso utilizados pelos hosts relevantes mostra que o registro `PTR` é o mais frequente em comparação aos registros `A` e `MX`.

Tabela 3. Resumo dos hosts relevantes encontrados para análise.

	DITL 2008	DITL 2009	DITL 2010
Total de Consultas	9.985.841	23.190.993	25.193.658
Total de Hosts Únicos	263.036	351.353	342.045
Total de Hosts Relevantes	138.558	178.519	306.124
Dist. Freq. - A|PTR|MX	32% | 53% | 12%	20% | 74% | 4%	39% | 43% | 16%

Para entender o padrão de comunicação entre os hosts identificados como relevantes, a Figura 3 apresenta o grau de saída e o grau de entrada dos vértices analisados em 2008, 2009 e 2010. Por razões de espaço, 20 mil hosts de cada ano foram escolhidos aleatoriamente e plotados. É possível destacar que muitos hosts são observados apenas enviando ou recebendo solicitações, visto que, os valores assumidos estão concentrados nos eixos X ou Y. De maneira geral, em 2008 e 2009, o grau de saída dos hosts está abaixo de 5 mil consultas enviadas, enquanto em 2010, esse padrão é divergente. Os valores identificados mostram que 22 hosts enviaram mais de 60 mil requisições e, por isso, foram investigados. Tais hosts estão distribuídos ao longo de sistemas autônomos de provedores de serviço de banda larga e provedores de Internet. O número AS foi obtido através do mapeamento `IP-to-ASN` disponível no projeto *Team CYMRU*.[3]

Para o grau de entrada, a maioria dos hosts recebeu menos de 600 consultas durante T, embora em 2010 seja possível observar hosts divergindo desse padrão. Uma análise superficial de tais hosts mostra que os nomes de domínios são consultados a partir seu nome reverso (`host.in-addr.arpa`), e que esses endereços correspondem a 78% de endereços de banda larga.

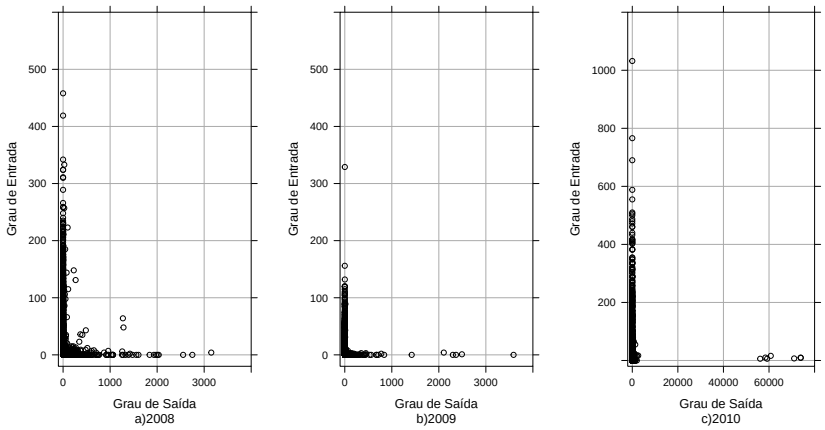


Figura 3. Visão geral das consultas enviadas ou recebidas na base de dados DITL 2008, 2009 e 2010.

[3]http://www.team-cymru.org/Services/ip-to-asn.html

Para ilustrar o comportamento dos hosts que enviaram mais de 60 mil consultas, a Tabela 4 apresenta o resumo das principais características dos 5 primeiros hosts desse grupo. Nessa tabela são apresentados o grau de entrada, grau de saída, razão, frequência dos registros recebidos e frequência dos registros enviados, respectivamente. A frequência dos registros de recursos de entrada denotam 100% de consultas do tipo `PTR`, com exceção do host de número 4, o qual recebeu consultas `PTR` (92.4%) e do tipo $\star$ (7.6%) [4]. As frequências dos registros de saída mostram os valores dos registros do tipo `A`, `PTR` e `MX`, entre os quais o tipo `A` é o mais frequente.

Tabela 4. Hosts relevantes em 2010 que enviaram mais de 60 mil consultas DNS.

Host	deg^-	deg^+	Razão	Freq. RR Entrada	Freq. RR Saída
1	50	204.616	4092.32	100%	75.2%| 14.7%| 14.4%
2	45	177.259	3930.08	100%	73.6%| 10.8%| 15.5%
3	37	130.205	3519.05	100%	70.3%| 9.4%| 19.8%
4	13	105.763	8135.61	92.4% | 7.6%	74.4%| 6.6%| 18.8%
5	37	97.858	2644.81	100%	81.1%| 6.9%| 11.8%

4.2. Classe Primária - Grau de Entrada

Para ilustrar como os hosts são observados nas classes de consultas, a classe primária do grau de entrada é demonstrada a seguir. Os 100 primeiros hosts que tiveram maior incidência de consultas foram identificados e atribuídos em intervalos distintos das classes de comportamentos. A Figura 4 apresenta a frequência total dos hosts em cada classe. Em 2008 e 2009, praticamente 100% dos hosts estavam localizados abaixo de 1000 consultas. Entretanto, em 2010, esse valor representa 60%. Devido à base de 2010 ser a mais recente, os valores encontrados serão descritos a seguir.

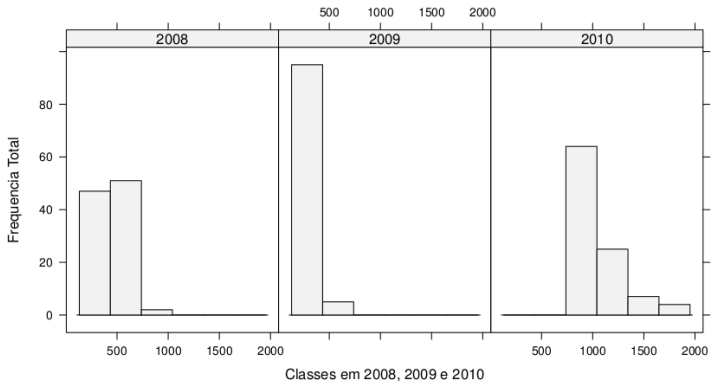


Figura 4. Intervalo de Classes para cada ano da base de dados.

Em 2010, 60% dos hosts que possuem maior incidência de consultas estão dentro do intervalo entre 500 e 1000 consultas. Nesse intervalo de classe, 83% dos hosts possuem o nome reverso apontando para endereços IPs e sistemas autônomos de banda larga. Os demais endereços são nomes no formato reverso que possuem o domínio inexistente (`NXDomain`). Do conjunto de hosts identificados nesse intervalo, vale destacar dois grupos; i) o conjunto de hosts que apenas recebem consultas; e o ii) conjunto de hosts que

[4]Consultas do tipo $\star$ são utilizadas para obter informações de todos os registros disponíveis do nome de domínio.

enviam e recebem consultas. O primeiro grupo representa 70% do total de hosts analisados. Tais hosts são consultados através dos registros de recurso `PTR`, $\star$ e `CNAME`[5]. O grau de saída desse grupo denota valores entre 0 e 1 consulta, os quais podem resultar na Razão igual a zero ou próxima de zero, respectivamente.

O segundo grupo representa 30% do intervalo citado e a frequência dos registros de recursos de saída denota comportamento suspeito. Por exemplo, os hosts identificados nesse grupo abusam do registro de recurso do tipo `MX`. Em média, esse registro representa 79% do total de consultas enviadas, enquanto os registros do tipo `A` e `PTR` representam 20% e 1%, respectivamente. Além disso, em comparação com o grau de saída (deg^+) dos hosts da Seção 4.1, os hosts do grupo ii apresentam volume baixo de consultas enviadas, em média esses hosts enviaram 231 solicitações DNS, das quais o tipo `MX` era predominante. A seguir é realizado uma análise passiva dos hosts encontrados nos conjuntos i e ii.

4.2.1. Análise Passiva dos Hosts Relevantes

Para validar a relevância dos hosts identificados nas Classes de Consultas, a análise passiva pode ser utilizada como uma alternativa. Informações adicionais sobre o vértice são utilizadas para determinar se o comportamento é benigno ou malicioso. Por exemplo, através da análise léxica do nome de domínio, um endereço IP é identificado como banda larga caso o nome de domínio seja composto por palavras como `cpe`, `vivax`, `adsl`, `modem`, `cliente`, `dial-up`, `dyn`, `dynamic`, `dsl`, `brasiltelecom`, `gvt`, `velox`, `user.vivozap` e `virtua`. Diversos trabalhos demonstram que clientes de banda larga - usuários domésticos - são as principais vítimas de ataques [Dagon et al. 2007, Antonakakis et al. 2010, Choi e Lee 2012].

De maneira semelhante, através da análise léxica, um servidor de email ou servidor de nomes é identificado caso o nome de domínio apresente palavras como `mail`,`exchange`, `pop`,`pop3`, `imap` ou `ns`, `dns` e `nameserver`, respectivamente. Desta forma, este trabalho assume como comportamento suspeito um endereço de banda larga abusando dos registros `PTR` ou `MX`, ou ainda, quando múltiplos servidores de email, dentro do mesmo espaço de tempo, consultam por um endereço de banda larga.

Para ilustrar o comportamento citado, considere os hosts relevantes que apenas receberam consultas, conforme descrição na Seção 4.2. Do total de 70 hosts, 88.5% são endereços de banda larga e os demais denotam nome de domínio inexistente (`NXDomain`). Uma verificação superficial mostrou que 85.8% dos hosts estavam cadastrados em listas negras por não seguirem as definições da RFC 2142 (1997). Tal RFC prevê um conjunto de endereços de emails especiais que um domínio deve possuir para atender reclamações de abuso ou de entrega de email. Além disso, a RFC 2142 encoraja que servidores de email implementem pelo menos uma caixa de correio capaz de lidar com os problemas citados [Crocker 1997]. Isto é, essa RFC estipula diretrizes de como um serviço de email deve operar corretamente na Internet. No entanto, para que os hosts sejam reconhecidos como servidores de email, o registro reverso deveria apontar para um nome de domínio diferente do nome de provedores de banda larga, caso contrário a resolução reversa não estaria seguindo a RFC 1912 [Barr 1996].

Adicionalmente, vale destacar que esses hosts possuem padrões semelhantes de

[5]É usado para especificar que um nome de domínio usa o endereço IP de outro domínio.

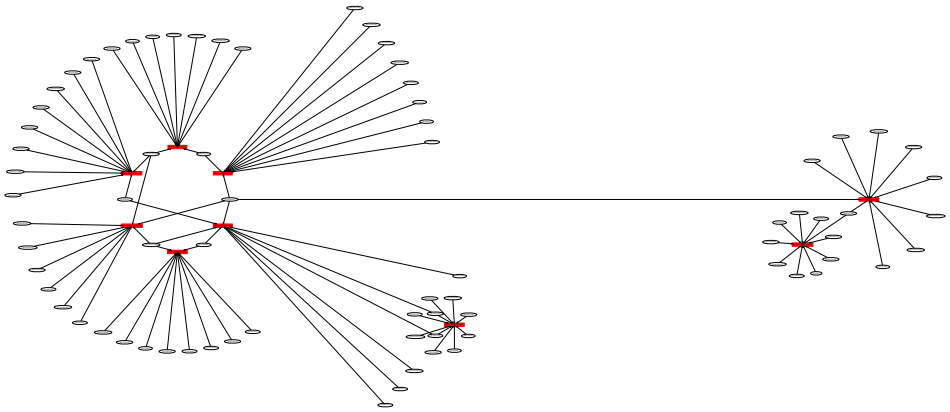


Figura 5. Padrão de consultas recebidas por *bots* para envio de spam. Quadrados vermelhos denotam nome da consulta, enquanto as elipses representam o IP de origem.

consultas recebidas. Primeiro, a maioria dos endereços IPs de origem que consultaram esses hosts são servidores de email e DNS. Segundo, o grau de entrada durante $\alpha = 10s$, tal que $\alpha \in T$, denota que esses hosts receberam em média 6 consultas a partir de endereços IPs distintos. Finalmente, é possível observar que um conjunto de hosts relevantes são consultados por um ou mais endereços IP de origem. Por questões de espaço, uma pequena parte desse padrão de comunicação é ilustrada na Figura 5. Nós vermelhos denotam os hosts frequentemente consultados e os nós em elipse representam os servidores de email ou de DNS. Diante desse contexto, é possível assumir que os hosts relevantes estão sendo consultados e validados após o envio em massa de spam.

```
01 - 201.b.c.x2z <domínio>.com.br MX
02 - 201.b.c.x2z ns1.<domínio>.com.br A
03 - 201.b.c.x2z itajuba.com.br MX
04 - 201.b.c.x2z <domínio>.com.br MX
05 - 201.b.c.x2z <domínio>.com.br MX
...
06 - 200.202.252.2 d.c.b.a.in-addr.arpa PTR
07 - 200.202.252.2 d.c.b.a.in-addr.arpa PTR
08 - 200.202.252.2 x2z.c.b.201.in-addr.arpa PTR
09 - 200.202.252.2 d.c.b.a.in-addr.arpa PTR
10 - 200.202.252.2 d.c.b.a.in-addr.arpa PTR
```

Figura 6. Exemplo de consultas realizadas por um *bot* para envio de spam.

Para comprovar o comportamento de envio de spam, a Figura 6 apresenta um exemplo dos hosts relevantes que enviaram e receberam consultas, conforme descrição na Seção 4.2. No exemplo é possível observar o endereço IP de origem, nome da consulta e o tipo de registro de recurso utilizado. As linhas 01-05 demonstram consultas do tipo `MX` e `A`. A linha 03 ilustra a consulta de serviço de correio eletrônico referente ao domínio `itajuba.com.br`. Em seguida, na linha 08, o endereço IP `x2z.c.b.201.in-addr.arpa` é solicitado pelo servidor de e-mail do domínio em questão (`200.202.252.2`). Tal comportamento pode ser identificado caso exista uma aresta entre o endereço IP do servidor de email e o endereço IP de origem inicial da consulta DNS. No entanto, para obter o endereço IP do servidor de email é necessário realizar uma nova consulta, a qual aumenta o tempo de análise dos hosts relevantes.

Além do padrão das consultas, outras características podem ser observadas nos

hosts relevantes. Tais máquinas enviam consultas com tempo de vida (TTL) dentro do intervalo [118,125], bit de fragmentação do IP (DF) ausente e marcam o bit de recursividade (RD). Em uma breve análise em ambiente virtualizado, é possível constatar que máquinas com Windows XP não marcam o bit DF nas consultas DNS, enquanto sistemas baseados em Unix habilitam esse bit. Computadores funcionando com o Windows também são identificados no tráfego DNS através do TTL indicado [Wessels e Fomenkov 2003]. A identificação de hosts usando Windows é útil para detectar máquinas comprometidas, pois esse sistema operacional possui o maior número de infecções.

Consultas recursivas (bit RD) não deveriam alcançar servidores de raiz, pois tais tipos de resoluções não são atendidas por servidores de primeiro nível [Castro et al. 2008]. Consultas recursivas também são resultado de aplicações maliciosas que podem utilizar sistemas de resolução independentes para evadir dos sensores de rede. Portanto, é possível assumir que tais máquinas estão utilizando o tráfego DNS para propagação de spam. A validação desse comportamento suspeito pode ser obtida através de consultas às listas negras, onde 93% dos hosts identificados ainda são encontrados nas bases de dados.

5. Conclusão

Neste trabalho foi apresentada uma metodologia para a identificação e classificação de anomalias de rede a partir da correlação das consultas no tráfego DNS. As consultas DNS foram modeladas em um grafo direcionado e o registro de recurso PTR utilizado para reforçar as conexões, resultando em subgrafos mais densos. Durante os experimentos foi identificado que os hosts que tinham maior incidência de consultas eram clientes de banda larga e que através da metodologia proposta, o volume de tráfego a ser analisado era em média 35% menor. A principal vantagem da metodologia proposta é a sua fácil implementação, pois apenas as solicitações DNS são utilizadas para indiciar comportamentos suspeitos.

Como trabalhos futuros, outros registros do tráfego DNS podem ser utilizados para identificar comportamentos suspeitos. Por exemplo, o tamanho das consultas utilizando o registro EDNS é útil para identificar ataques de amplificação de resposta DNS. Os registros CNAME e TXT podem ser utilizados para detecção de tunelamento de tráfego através do protocolo DNS. Finalmente, algoritmos de aprendizagem de máquina seriam utilizados para detectar e classificar comportamentos suspeitos em tempo real a partir do emprego das características extraídas dos hosts relevantes.

Agradecimentos

Este trabalho agradece DNS-OARC e ao *Registro.br* pelo acesso ao tráfego. Este trabalho foi desenvolvido com o apoio do Governo do Estado do Amazonas por meio Fundação de Amparo à Pesquisa do Estado do Amazonas, com a concessão de bolsa de estudo.

Referências

Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., e Feamster, N. (2010). Building a dynamic reputation system for dns. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 18–18, Berkeley, CA, USA. USENIX Association.

Barbosa, K. R. S. e Souto, E. (2009). Análise passiva do tráfego dns da internet brasileira. In *IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg 2009*, pages 203–216, Campinas.

Barr, D. (1996). RFC 1912: Common DNS operational e configuration errors. http://www.ietf.org/rfc/rfc1912.txt.

Castro, S., Wessels, D., Fomenkov, M., e Claffy, K. (2008). A day at the root of the internet. *ACM SIGCOMM Computer Communication Review (CCR)*, 38(5):41–46.

Choi, H. e Lee, H. (2012). Identifying botnets by capturing group activities in dns traffic. *Computer Networks*, 56(1):20–33.

Crocker, D. (1997). RFC 2142: Mailbox names for common services, roles e functions. http://www.ietf.org/rfc/rfc2142.txt.

Dagon, D., Gu, G., Lee, C., e Lee, W. (2007). A taxonomy of botnet structures. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*, pages 325 –339.

DITL (2014). A day in the life of the internet (ditl). `https://www.dns-oarc.net/oarc/data/ditl` (acessado em 01/03/2014).

Ishibashi, K., Toyono, T., Toyama, K., Ishino, M., Ohshima, H., e Mizukoshi, I. (2005). Detecting mass-mailing worm infected hosts by mining dns traffic data. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, MineNet '05, pages 159–164, New York, NY, USA. ACM.

Jiang, N., Cao, J., Jin, Y., Li, L., e Zhang, Z.-L. (2010). Identifying suspicious activities through dns failure graph analysis. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 144–153.

Kumagai, M., Musashi, Y., Romana, D., Takemori, K., Kubota, S., e Sugitani, K. (2010). Ssh dictionary attack and dns reverse resolution traffic in campus network. In *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*, pages 645–648.

Mockapetris, P. (1987). RFC 1034: Domain names - concepts and facilities. http://www.ietf.org/rfc/rfc1034.txt.

Ramachandran, A., Feamster, N., e Dagon, D. (2006). Revealing botnet membership using dnsbl counter-intelligence. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, SRUTI'06, pages 8–8, Berkeley, CA, USA. USENIX Association.

Shibata, N., Musashi, Y., Romana, D., Kubota, S., e Sugitani, K. (2012). Trends in host search attack in dns query request packet traffic. In *Intelligent Networks and Intelligent Systems (ICINIS), 2012 Fifth International Conference on*, pages 126–129.

Shin, S. e Gu, G. (2010). Conficker and beyond: A large-scale empirical study. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 151–160, New York, NY, USA. ACM.

Wessels, D. e Fomenkov, M. (2003). Wow, that's a lot of packets. In *Passive and Active Network Measurement Workshop (PAM)*, pages 1–9, San Diego, CA. PAM.

Yuchi, X., Wang, X., Li, X., e Yan, B. (2009). Dns measurements at the .cn tld servers. In *Proceedings of the 6th international conference on Fuzzy systems and knowledge discovery - Volume 7*, FSKD'09, pages 540–545, Piscataway, NJ, USA. IEEE Press.

Um Mecanismo Agregador de Atributos Mediado pelo Cliente Alinhado ao Programa de E-GOV.BR

Marcondes Maçaneiro[1,2], Fábio Zoz[2], Michelle Silva Wangham[1]

[1]Laboratório de Sistemas Embarcados e Distribuídos (LSED) –

Universidade do Vale do Itajaí (UNIVALI)

[2]Curso de Sistemas de Informação – Centro Universitário para o Desenvolvimento do Alto ale do Itajaí, UNIDAVI.

{marcondes, zozfabio}@unidavi.edu.br, wangham@univali.br

Abstract. *The use of multiple identity providers (IdPs) in IdM systems can bring benefits to users, especially regarding privacy of data. This paper describes an aggregation mechanism able to gather and to join users' attributes that are distributed in multiple IdPs. These attributes can be presented to providers that require attributes, which are not in a single IdP. The proposed mechanism is innovative in adopting a client-mediated approach, which makes use of an active client in the user's environment and follows the recommendations of the ePING architecture of Gov.br's Program. The implementation results and the use of proposed mechanisms demonstrate that it gets more flexibility to a Gov Federation and also assures privacy without prejudicing the interoperability of E.Gov applications.*

Resumo. *O uso de múltiplos provedores de identidades (IdPs) em sistemas de IdM pode trazer vantagens para os usuários, principalmente, para privacidade de seus dados. Este artigo descreve um mecanismo agregador de atributos capaz de coletar e unir os atributos dos usuários disponibilizados em múltiplos IdPs, para que estes possam ser apresentados para provedores que exigem atributos que não estão em um único IdP. O mecanismo proposto é inovador ao adotar uma abordagem mediada pelo cliente, que faz uso de um aplicativo executado no ambiente do usuário e que segue as recomendações da arquitetura ePING do Programa Gov.br. Os resultados obtidos com a implementação e uso do mecanismo proposto demonstram que este traz mais flexibilidade para um sistema federado e garante a privacidade dos usuários sem prejudicar a interoperabilidade do sistema e de uma aplicação de E.Gov.*

1. Introdução

Sistemas de gerenciamento de identidades (*Identity Management - IdM*) federadas permitem o compartilhamento dos atributos do usuário e a autenticação única através de múltiplos domínios, tornando-se facilitadores para os sistemas governamentais [Baldoni 2012]. Nos últimos anos, alguns governos aprovaram estratégias nacionais de gestão de identidades baseadas no modelo federado buscando melhorar seus serviços de governo eletrônico, dentre estes, destacam-se: Nova Zelândia, Austrália, Canadá e Estados Unidos [OECD 2011].

A maioria dos sistemas de IdM federadas restringe a fonte de identidades e de atributos a um único provedor de identidades (IdP), em qualquer sessão criada com um provedor de serviços (SP) [Chadwick e Inman 2013]. Com isto, as autorizações são limitadas a um subconjunto de atributos da identidade do usuário. Segundo Klingenstein (2007), os sistemas de IdM federadas são sólidos e garantem o acesso federado de seus usuários, porém, muitas vezes, questões referentes à privacidade dos usuários não são devidamente consideradas, já que durante as trocas de informações que ocorrem nesses sistemas, os provedores podem rastrear a identidade do usuário e seus acessos.

No contexto das aplicações de Governo Eletrônico, diante das diversas esferas governamentais, observa-se como comum em uma federação que um usuário possua atributos espalhados em múltiplos IdPs, cada qual mantendo apenas os atributos dos usuários que são de sua responsabilidade. Observa-se ainda que, para algumas aplicações, é necessário que estes sejam coletados. Esta união, muitas vezes processada por uma terceira parte confiável, é um procedimento conhecido como agregação de atributos [Hatakeyma e Shima 2008]. Nesta abordagem, a terceira parte mantém o controle das informações e acessos de usuário, o que pode comprometer a sua privacidade.

Este artigo tem por objetivo descrever um mecanismo agregador de atributos mediado pelo cliente que atende as recomendações da arquitetura ePING (Padrões de Interoperabilidade de Governo Eletrônico do Brasil) [Brasil 2014]. O mecanismo proposto tem como foco garantir a privacidade dos usuários e trazer mais flexibilidade a um sistema de IdM federadas governamental ao possibilitar a agregação de atributos de múltiplos IdPs que seguem o padrão SAML[1], por meio de um cliente ativo executado na máquina do usuário.

Este artigo está organizado em seis seções. A Seção 2 apresenta os principais conceitos envolvidos no problema e na solução proposta. A Seção 3 apresenta e compara os trabalhos relacionados selecionados a partir de uma revisão sistemática da literatura. Na Seção 4, as premissas e o detalhamento do funcionamento do mecanismo agregador de atributos proposto são descritos. Os resultados experimentais relativos à implementação do mecanismo e o seu uso em uma aplicação de e-Gov são discutidos na Seção 5. Por fim, na Seção 6, são apresentadas as considerações finais.

2. Gestão de Identidades Federadas

A gestão de identidades pode ser entendida como o conjunto de processos e tecnologias usados para garantir a identidade de uma entidade (usuário ou um dispositivo), garantir a qualidade das informações de uma identidade (identificadores, credenciais e atributos) e para prover procedimentos de autenticação, autorização e auditoria [ITU 2009]. As entidades envolvidas em um sistema de IdM são: (i) usuário ou dispositivo, entidade que utiliza um serviço fornecido por um provedor de serviços; (ii) provedor de identidades (*Identity Provider* – IdP), responsável por manter a base de dados de usuários do domínio e validar suas credenciais (autenticar usuários); e (iii) provedor de serviços (*Service Provider* – SP), que oferece recursos ou serviços aos usuários.

[1] SAML é o padrão recomendado pela arquitetura E-PING e é o mais adotado por países que seguem o modelo de IdM federadas [OECD 2011][Brasil 2014] .

Dentre os modelos de IdM, no contexto de Governo Eletrônico, destaca-se o federado [OECD 2011]. Neste modelo, a tarefa de autenticação é realizada a partir de múltiplos provedores de identidades, que participam de um círculo de confiança entre diferentes domínios administrativos. Um domínio administrativo pode representar, por exemplo, uma empresa, uma universidade ou uma unidade governamental. Este domínio administrativo é composto de usuários, provedores de serviços (SPs) e um provedor de identidades (IdP). O gerenciamento de identidades federadas possibilita o compartilhamento de atributos do usuário, além da autenticação única[2] (SSO – *Single Sign On*) através de múltiplos domínios [Landau et al 2009].

Segundo [Chadwick e Inman 2013], a maioria dos sistemas de IdM federadas limita o usuário para que este escolha apenas um provedor de identidades por sessão. A introdução de mecanismos agregadores de atributos permite que, de forma segura, usuários possam acessar diferentes IdPs, o que possibilita a agregação de atributos de múltiplas fontes, sem a necessidade do usuário se autenticar separadamente em cada IdP (autenticação SSO nos IdPs). Para atender ao requisito de privacidade dos usuários, um mecanismo agregador de atributos deve inviabilizar o rastreamento das ações dos usuários e dos seus atributos de identidade [Chadwick e Inman 2013].

Na tentativa de evitar o comprometimento das informações do usuário, alguns trabalhos descritos na literatura buscam resolver o problema referente à privacidade dos usuários, por meio de soluções de IdM federadas centradas no usuário, que visam atribuir o controle das informações aos próprios usuários, já que estes são os mais habilitados a liberar os atributos em suas contas nos IdPs. Segundo [Hoellrigl et al. 2010], uma característica importante sobre os sistemas de IdM centrados nos usuários é a capacidade do usuário de poder escolher o IdP que deseja utilizar.

De acordo com [Klingenstein 2007], existem seis abordagens para implementação de mecanismos de agregação de atributos. A primeira assume que o usuário tenha um identificador único comum entre todas as autoridades de atributos. Este identificador é um nome X.500 contido em um certificado X.509, emitido por uma autoridade certificadora [Landau *et al.* 2009]. A segunda abordagem é classificada como *proxy* de identidades. Um IdP *proxy* transforma ou estende uma identidade obtida por um IdP em uma identidade que contém as informações necessárias ao SP. O IdP *proxy*, neste caso, deve ser confiável tanto para o SP que depende deste, como também pelos IdPs que disponibilizam os atributos ao IdP *proxy*. O IdP *proxy* é capaz de identificar todos os atributos sobre qualquer identidade [Chadwick et al 2010].

A terceira abordagem é considerada uma especificação de um *proxy*. Esta abordagem é denominada de proxy de retransmissão de identidade, embora as trocas de atributos no modo de retransmissão de identidade sejam semelhantes ao modelo de IdP *proxy*, neste modelo tem-se a garantia da retransmissão das asserções de atributos. Neste caso, o IdP *proxy* não irá assinar as asserções de atributos, mas sim retransmiti-las no formato original (assinadas pelo IdP original). O SP irá receber as asserções de atributos criptografadas [Chadwick et al 2010].

[2] Na autenticação única (SSO), o usuário autentica-se uma única vez em seu IdP de origem e pode acessar diferentes SPs. No caso da autenticação federada, estes SPs podem estar outros domínios administrativos.

A abordagem de agregação de atributos mediada pelo cliente faz uso de clientes inteligentes (clientes ativos) para criar solicitações de atributos aos múltiplos IdPs. Exemplos destes clientes são o ECP SAML (do inglês *Enhanced Client or Proxy*) e os clientes ativos do *CardSpace* [Klingenstein 2007]. A quinta abordagem é a federação de identidades que depende da capacidade do usuário em associar as identidades que controla fazendo a ligação entre os IdP. Um agente do usuário autenticado com sucesso em dois IdPs diferentes pode controlar ambos os IdPs e criar um identificador unidirecional persistente, permitindo que um IdP aponte para a identidade relacionada no segundo IdP. Por fim, a última abordagem possível é a mediada pelo SP. Apesar da autenticação do usuário ser única para cada IdP, o SP necessita realizar excessivos redirecionamentos para completar a agregação dos atributos [Klingenstein 2007].

3. Trabalhos Relacionados

Após a execução de um protocolo de busca (revisão sistemática), foram identificados nove trabalhos (entre 2008 e 2013) que tratam do problema da agregação de atributos e descrevem mecanismos agregadores.

Em [Lee *et al.* 2008], um modelo de agregação de atributos, que segue a abordagem de mediada pelo provedor de serviços (SP) e que utiliza a avaliação da reputação dos IdPs no processo de agregação dos atributos do usuário, é descrito. Para auxiliar a agregação, cada SP obtém os atributos dos IdPs e os armazena em seu repositório de dados local. Esta solução é simples de ser desenvolvida, mas possibilita o rastreamento das informações dos usuários por parte dos SPs, visto que este irá mediar todo o processo de agregação de atributos e viola a privacidade, ao armazenar no SP os atributos do usuário.

O trabalho proposto por [Hatakeyama e Shima 2008] apresenta uma infraestrutura para a gestão de privilégios em uma federação a fim de vincular todos os tipos de perfis dos usuários. O modelo permite que os usuários conectem suas contas de outros provedores e consigam gerenciar tais contas de forma centralizada. A proposta dos autores está na criação de um novo módulo de segurança (do inglês *Trust Server Provider* – TSP), que segue uma abordagem de *proxy* e que possibilita o rastreamento das informações dos usuários.

Em [Chadwick *et al.* 2010], é descrito um mecanismo agregador de atributos que segue a abordagem baseada em proxy e que permite que SPs autorizem solicitações de acesso dos usuários com base em atributos afirmados por vários provedores de identidade (IdP). O modelo utiliza um componente chamado de serviço de ligação (do inglês – *Linking Service* - LS), que tem por objetivo ligar as contas dos usuários que estão em diferentes IdPs. A solução proposta está baseada no protocolo SAML 2.0, porém, por se tratar de uma solução de proxy, possui as desvantagens em relação à privacidade apontadas anteriormente.

O modelo proposto por [Hoellrigl *et al*. 2010] apresenta um mecanismo de delegação de identidades controlado pelo usuário. Este delegado de identidade atua em nome do usuário quando um SP deseja acessar seus atributos, mesmo que o usuário não esteja conectado. Apesar de seguir uma abordagem centrada no usuário, o mecanismo implementa uma abordagem de agregação de retransmissão, pois este é responsável por

mediar todas as requisições do usuário, porém não armazena os atributos e não está no ambiente computacional do cliente (é uma aplicação Web).

A proposta de [Vossaert *et al.* 2010] define uma abordagem para um sistema de gestão de identidades centrada no usuário, que aborda a privacidade dos usuários. No modelo de agregação de atributos proposto pelos autores, alguns atributos do usuário podem estar disponíveis no cache do elemento seguro temporariamente. No modelo, tem-se um módulo de segurança (*Trusted Module* - TM) para cada usuário, que é representado por um cartão inteligente (*smart card*) que fica de posse do usuário. Esse cartão permite que os usuários possam interagir com o sistema. Esta proposta, por ser centrada no usuário e mediada pelo cliente, favorece a garantia de privacidade dos usuários, mas é apenas uma abordagem conceitual que não foi implementada e avaliada. Além disso, a solução não adota nenhum padrão de interoperabilidade para lidar com identidades de IdPs heterogêneos (solução proprietária implementada no *smartcard*).

O trabalho proposto por [Hulsebosh *et al*. 2011], apresenta uma plataforma de colaboração virtual (do inglês *Virtual Collaboration Platform* - VCP) para a *SURFnet*. A plataforma aproveita a infraestrutura da federação acadêmica já existente, que se baseia no padrão *SAML2*. Esta plataforma implementa um mecanismo agregador de atributos no formato de *proxy* ou retransmissão, conforme desejado.

[Chadwick et al 2011] estendem o seu trabalho anterior para integrar ao projeto *Logins4Life* que visa o uso das contas sociais dos usuários da comunidade acadêmica para obter acesso às ferramentas disponibilizadas pelas instituições. No mecanismo proposto, os autores definiram um provedor de serviço confiável (*Trusted Service Provider* - *TSP*) responsável por ligar as diferentes contas dos usuários. O TSP visa ainda prover segurança à base de dados com os atributos do usuário e contêm um *proxyIdP*. Semelhante ao trabalho anterior, este trabalho segue a abordagem baseada em *proxy* e possibilita ainda o armazenamento de atributos dos usuários (de diferentes IdPs), o que prejudica o requisito de privacidade dos usuários.

Com objetivo de aprimorar a preservação de privacidade no processo de agregação de atributos, [Chadwick et al 2003] definiram o *Trusted Attribute Aggregation Service – TAAS*, responsável por coletar e agregar os atributos de um usuário a ser entregue para o SP. Ao contrário dos trabalhos anteriores, este serviço não armazena os atributos dos usuários e, de forma semelhante, a um seletor de identidades do CardSpace permite que um usuário selecione quais atributos de quais IdPs serão encaminhados a um dado SP. A diferença está que o TAAS não é um cliente ativo, mas sim uma aplicação web (*proxy*). Outra característica desta solução é que o usuário precisará se autenticar em cada IdP para que este emita a asserção SAML com seus atributos para evitar a rastreabilidade dos atributos do usuário por parte de um IdP.

A Tabela 1 apresenta um resumo comparativo dos trabalhos relacionados considerando as seguintes características: (1) se a proposta foi implementada e avaliada, (2) as tecnologias de IdM adotadas na definição da proposta, (3) o padrão de interoperabilidade utilizado e (4) qual a abordagem de agregação de atributos que a proposta implementa. Dentre os modelos e mecanismos de agregação de atributos apresentados, a abordagem de *proxy* de identidades se mostrou a mais comum.

Como visto anteriormente, a abordagem mediada pelo cliente busca evidenciar a privacidade dos usuários no uso do mecanismo agregador [Klingenstein, 2007]. A

justificativa para a pouca adoção desta abordagem se deve à dificuldade de prover a autenticação SSO, uma vez que nas soluções citadas que fazem uso desta abordagem, o usuário deve se autenticar várias vezes em diferentes provedores de identidades. Somente o trabalho de [Vossaert *et al* 2010] apresenta uma proposta que segue a abordagem mediada pelo cliente (faz uso de um cliente ativo) baseada no uso de *smartcards*, porém, esta proposta não foi implementada e avaliada. O mecanismo agregador proposto neste trabalho e o seu diferencial diante das soluções apresentadas estão indicados na Tabela 1 e descritos nas próximas seções.

Tabela 1: Tabela comparativa dos Trabalhos Relacionados

TRABALHOS RELACION.	IMPL.	TECNOLOGIAS DE IDM ADOTADAS	PADRÃO DE INTEROPERABILID.	ABORDAGEM DA AGREGAÇÃO DE ATRIBUTOS
Lee et al. (2008)	Não	Apenas modelo conceitual	XML	Mediado pelo SP
Hatakeyama e Shima (2008)	Não	OpenID, CardSpace, SAML, OAuth	SAML	Proxy
Chadwick *et al.* (2010)	Sim	SAML	SAML	Proxy
Hoellrigl *et al.* (2010)	Sim	CardSpace, Active Directory Federation Services	WS-Trust	Retransmissão
Vossaert et al. (2010)	Não	OpenID, Shibboleth, CardSpace	-	Mediado pelo Cliente
Chadwick et al. (2010)	Sim	SAML, OpenID, OAuth	SAML	Proxy
Hulsebosch (2011)	Sim	SAML, OAuth	SAML, Protocolo *OpenSocial*	Proxy ou Retransmissão
Chadwick et al (2011)	Sim	SAML, Facebook Connect, OAuth, OpenID	SAML	Proxy
Chadwick et al (2013)	Sim	SAML	XML e SAML	Retransmissão
Modelo Proposto	**Sim**	**SAML**	**SAML e Schemas XML**	**Mediado pelo Cliente**

4. Mecanismo Agregador de Atributos Baseado em Cliente Ativo

O governo brasileiro ainda não definiu a estratégia nacional de gestão de identidades a ser adotada nas aplicações de Governo Eletrônico. O que existe é apenas a arquitetura ePING, que traz algumas diretrizes para definição de estratégias de interoperabilidade entre sistemas [Brasil 2014]. Dentre estas, destacam-se o uso do SAML como padrão para a troca de informação sobre autenticação e autorização entre domínios, da especificação WS-Security 1.1 para o fornecimento de segurança às mensagens trocadas e WS-Trust 1.4 para a gestão de relacionamentos confiáveis (intermediação).

Para conceber uma estratégia nacional de gestão de identidades federadas para o governo brasileiro, um dos problemas que precisa ser solucionado é a agregação de atributos de identidades de um usuário (cidadão). Poder agregar de forma segura atributos que estão distribuídos em diferentes IdPs proverá uma maior flexibilidade a esta estratégia. É comum um cidadão possuir atributos de identidade distribuídos em diferentes provedores, tais como os do DETRAN, polícia federal, receita federal, sistema único de saúde entre outros. Alguns serviços governamentais, como a emissão de passaportes, exigem que um usuário apresente atributos que podem estar distribuídos

em múltiplos IdPs. Este trabalho descreve um mecanismo agregador de atributos mediado pelo cliente que atende as recomendações da arquitetura ePING, que garanta a privacidade dos usuários e que visa trazer mais flexibilidade a um sistema de gerenciamento de identidades federadas governamental.

Visando impedir a rastreabilidade dos atributos de identidade do usuário, possível nas soluções baseadas em *proxy*, o mecanismo agregador proposto segue a abordagem mediada pelo cliente, com o diferencial de ser alinhado à arquitetura ePING. O procedimento de agregação é executado por um aplicativo no ambiente operacional do usuário, chamado de cliente ativo. O mecanismo agregador gerencia ainda pseudônimos, conforme definido na especificação SAML, para aumentar a privacidade do usuário e dificultar o rastreamento de suas informações nos SPs.

O mecanismo agregador proposto assume algumas premissas. A primeira está na existência de uma federação governamental. Esta federação deve envolver todas as esferas governamentais, tais como: governos federal, estadual e municipal. Nesta federação, existem provedores de identidades (IdPs) e provedores de serviços governamentais (SPs) que possuem relações de confiança e que estão em conformidade com a especificação SAML. E, por fim, considera-se como premissa que o código do mecanismo agregador foi homologado e assinado por uma entidade confiável da federação. IdPs e SPs devem ainda seguir outras diretrizes de interoperabilidade da arquitetura ePING, como o uso de Serviços *Web RESTful*, o uso do protocolo SSL e o padrão XML (como linguagem de intercâmbio de dados).

Dentro da federação governamental considerada, um usuário (cidadão) tem atributos distribuídos em múltiplos IdPs. Um SP que oferece serviços governamentais via Internet pode requerer um subconjunto desses atributos de um usuário para prover determinados serviços. Para coletar esse subconjunto de atributos, utiliza-se do mecanismo agregador de atributos, que em nome e com a aprovação do usuário coleta os atributos em diversos IdPs e os compartilha com o SP. O mecanismo agregador não compartilha qualquer informação sem o consentimento do usuário.

A Figura 1 ilustra a visão geral do mecanismo agregador proposto. Alguns novos serviços foram definidos para integração do mecanismo agregador de atributos aos serviços de uma federação. O SDPCA, Serviço de Descoberta de Provedor de Cliente Ativo, é responsável por apresentar ao usuário uma lista de provedores de aplicativos de cliente ativo[3] homologados pelo governo. Caberá ao usuário indicar o provedor de cliente ativo (PCA) em que confia para fazer o download do aplicativo. É importante destacar que tanto o SDPCA, quanto os PCAs devem ser SPs da federação (estão no círculo de confiança da federação). Em todas as trocas, utiliza-se o protocolo SSL para estabelecer um canal seguro de comunicação entre as entidades envolvidas.

No passo 1 da Figura 1, o usuário, por intermédio de seu navegador Web, tenta acessar um serviço da federação. Por ser um serviço que exige autenticação, o navegador do usuário é redirecionado para o IdP indicado pelo serviço para proceder com a autenticação (passo 2). Após usuário informar os dados para a autenticação (passo 2a), o IdP autentica o usuário, emite uma asserção de atributos para este (passo

[3] O software de cliente ativo pode ser desenvolvido por órgãos do governo ou também por empresas privadas, porém, estes precisarão passar por um processo de homologação.

2b) e redireciona o navegador para o SP (passo 3a). Para concretizar a ação solicitada pelo usuário, o SP indica quais atributos do usuário este necessita (passo 3b). Neste momento, o usuário deve confirmar que deseja prosseguir com o processo de agregação de atributos. Para obter o software do cliente ativo, responsável pela agregação (passo 4), o navegador do usuário é redirecionado para o SDPCA mantido pela federação.

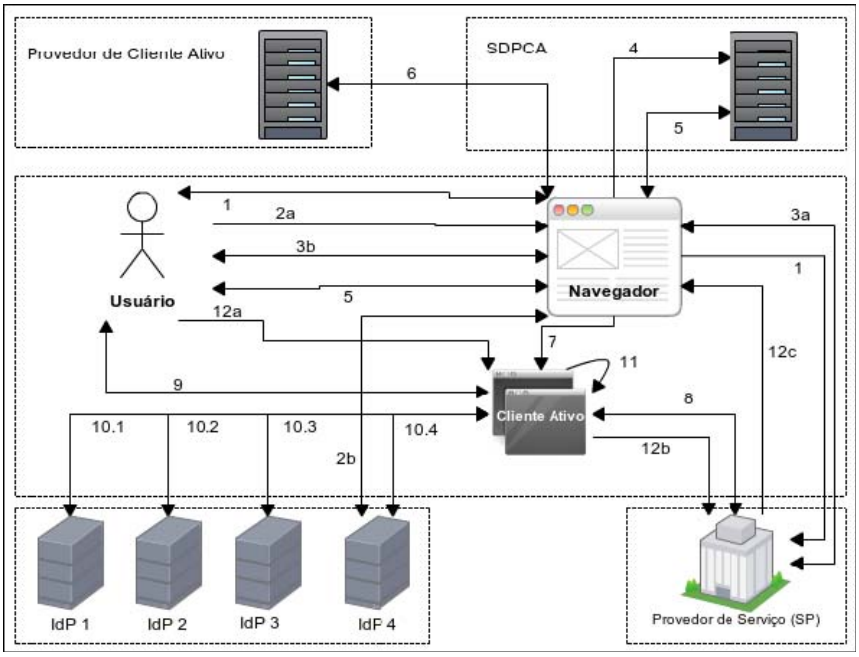


Figura 1- Visão geral do mecanismo agregador de atributos proposto.

Após o usuário selecionar um dos PCAs (passo 5), o navegador do usuário é redirecionado para o provedor selecionado para fazer o *download* da aplicação de cliente ativo (passo 6). Após o *download* do cliente ativo, este é executado no ambiente operacional do usuário (passo 7). Para efetuar a agregação de atributos, o cliente ativo deve solicitar ao SP a asserção assinada que indica os atributos necessários (passo 8). Os atributos necessários são apresentados ao usuário para que este indique quais IdPs deseja utilizar para cada atributo (passo 9). Como a autenticação SSO é garantida, o usuário precisa se autenticar, via cliente ativo, apenas no primeiro IdP indicado (passo 10.1), os demais aceitam o *token* de autenticação emitido pelo IdP e respondem à solicitação de atributos (passo 10.2, 10.3 e 10.4).

No passo 11, o cliente ativo efetua a agregação de atributos. No passo 12a, o cliente ativo solicita que o usuário confirme a liberação dos atributos e então encaminha (passo 12b) os atributos agregados para o SP. De forma semelhante a abordagem de retransmissão, o cliente ativo não irá assinar as asserções de atributos, mas sim retransmiti-las no formato original (assinadas pelo IdP original). Logo, qualquer violação das asserções recebidas feita na máquina do usuário[4] pode ser detectada pelo SP. O SP de posse dos atributos enviados (passo 12c) poderá permitir ou não o acesso ao serviço solicitado pelo usuário.

Nos passos da autenticação do usuário (passos 2 e 10), recomenda-se que os IdPs utilizem métodos de autenticação que evitem a adivinhação de senhas (ataque de

[4] Considerando que o ambiente de execução do usuário possa ser malicioso.

força bruta). Recomenda-se o uso de mecanismos de autenticação mais fortes que os baseados em senha, mas que podem ser utilizados implementados via cliente ativo. Por exemplo, certificados digitais ou autenticação de dois fatores que combinem uma prova de posse (e.g. celular). Para algumas aplicações governamentais, o uso de uma identidade digital (*smartcard*) ou o novo RIC (Registro de Identidade Civil Brasileiro) pode ser exigido para prover esta autenticação forte baseada em certificados digitais.

O processo de agregação de atributos pode ser classificado como dinâmico ou estático e transitório ou permanente [Chadwick et al. 2010]. O mecanismo agregador proposto oferece três modos de funcionamento. O primeiro modo é o transitório dinâmico. Neste modo, não há uma política de liberação de atributos pré-definida para os SPs (dinâmico) e o usuário precisará se autenticar em cada IdP que solicitar atributos (transitório). Neste modo, a autenticação SSO nos IdPs[5] não é suportada.

O segundo modo é o permanente dinâmico. Neste modo, ainda não há uma política de liberação de atributos pré-definida para os SPs (dinâmico). Além disso, neste modo, a autenticação SSO é garantida para solicitar atributos em múltiplos IdPs (permanente). Neste caso, o *token* de autenticação do primeiro IdP é compartilhado e aceito nos demais IdPs. E, por fim, o terceiro modo que é o permanente estático. Neste modo, o usuário pode criar uma política de liberação de atributos após o processo de agregação, assim como pode salvar informações sobre quais escolhas de IdP foram feitas pelo cliente ativo. Além disso, a autenticação SSO nos IdPs é garantida. Todos os modos exigem o consentimento do usuário para liberação de atributos.

Um padrão para as trocas de mensagens entre os provedores de serviço e o Cliente Ativo foi definido com estruturas de esquemas de dados XML (*XML Schemas*). Estes esquemas padronizam as requisições de atributos enviadas ao cliente Ativo (ver exemplo na Figura 2.a) e também a resposta da agregação de atributos com as asserções SAML (ver exemplo na Figura 2.b) que devem sem compartilhadas com os SPs.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <SAMLAgregator>
3    <SAMLRequest>
4      <attribute>CPF</attribute>
5    </SAMLRequest>
6    <SAMLRequest>
7      <attribute>TITULOELEITOR</attribute>
8    </SAMLRequest>
9    <SAMLRequest>
10     <attribute>RG</attribute>
11   </SAMLRequest>
12 </SAMLAgregator>
```

a. *XML Request*

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <SAMLAgregator>
3    <SAMLResponse>
4      <attribute>CPF</attribute>
5      <SAML>SAML...SAML</SAML>
6    </SAMLResponse>
7    <SAMLResponse>
8      <attribute>RG</attribute>
9      <SAML>SAML...SAML</SAML>
10   </SAMLResponse>
11   <SAMLResponse>
12     <attribute>TITULOELEITOR</attribute>
13     <SAML>SAML...SAML</SAML>
14   </SAMLResponse>
15 </SAMLAgregator>
```

b. *XML Reply*

Figura 2- Exemplos de um pedido e de uma resposta de atributos

5. Implementação e Resultados

Com o objetivo de avaliar (1) a flexibilidade proporcionada com o uso do mecanismo agregador de atributos proposto, (2) o impacto sobre a interoperabilidade do sistema de

[5] Vale ressaltar que a autenticação SSO nos SPs é garantida.

gerenciamento de identidades adotado, (3) a privacidade e a (4) usabilidade dos usuários ao fazer uso do cliente ativo, um protótipo do mecanismo agregador de atributos foi desenvolvido e integrado a um cenário fictício de solicitação de emissão de passaporte.

O processo de emissão de passaporte é demorado e burocrático. A Polícia Federal exige a apresentação de alguns documentos que comprovem a identidade do cidadão, tais como: Registro de Geral (RG), Título de Eleitor, Documento de Cadastro de Pessoa Física (CPF). Após efetuar o cadastro das informações e pagar a taxa do serviço, o cidadão deve apresentar seus documentos originais em uma agência de atendimento, para então, recolher as digitais e fotografia do cidadão. Este procedimento não evita a apresentação de documentos falsificados. Pode-se tornar este procedimento automático e mais seguro se os atributos do usuário forem coletados diretamente nos IdPs da Federação Governamental com o uso do mecanismo agregador de atributos.

O uso do mecanismo agregador, neste cenário, contemplou os seguintes passos: (1) o usuário, por meio do navegador Web, acessa o SP da Polícia Federal (PF). Como o serviço exige que o usuário se autentique, o navegador é redirecionado para o IdP de confiança do SP da PF; (2) o usuário se autentica no provedor de identidades (login e senha) e o seu navegador é redirecionado de volta para o SP da PF; (3) após o usuário indicar que consente com o processo de agregação de atributos, o navegador é redirecionado para o provedor SDPCA; (4) o usuário seleciona o provedor para realizar o *download* do cliente ativo e o seu navegador é redirecionado para o site do provedor escolhido; (5) o usuário efetua o download da aplicação e executa o aplicativo de cliente ativo em sua máquina. Em seguida, o cliente ativo obtém a lista de atributos requeridos pela aplicação (conforme Figura 2.a); (6) o cliente ativo apresenta os atributos e solicita que o usuário indique quais IdPs contêm os atributos requeridos; (7) após indicar os IdPs para cada atributo, o usuário se autentica em cada IdP para que o cliente ativo recolha e reúna as asserções de atributos; e (8) por fim, o cliente ativo agrega as asserções SAML (sem modificá-las) e as envia para o SP da polícia federal (Figura 2.b).

Para o desenvolvimento do protótipo de cliente ativo, foi escolhida a plataforma Java por oferecer portabilidade e por oferecer uma solução robusta para o desenvolvimento de códigos de execução remota (aplicativos *Java Web Start* - JWS). A tecnologia JWS permite que um aplicativo, que está disponível em um provedor, seja migrado para a máquina do usuário e seja executado em seu ambiente operacional[6]. Vale destacar ainda que o aplicativo JWS é assinado digitalmente pelo PCA. Os provedores de serviços SPDCA, PCA e o serviço para emissão de passaporte que foram utilizados nos experimentos de avaliação do mecanismo proposto foram desenvolvidos em PHP e o Apache foi o servidor Web utilizado. Todos os SPs fazem uso de certificados digitais SSL. Os IdPs utilizados nos experimentos estão de acordo com a especificação SAML2 e foram implementados utilizado o *framework simpleSAMLPHP*.

Após a execução de testes de software funcionais e não funcionais (incluindo os de segurança), realizou-se um experimento que envolveu especialistas que trabalham ou prestam serviços para o governo e a aplicação. Estes especialistas, após o uso do protótipo, responderam uma pesquisa de satisfação (teste de usabilidade). Nesta

[6] O *Java Web Start* é iniciado automaticamente quando é feito o primeiro *download* do aplicativo Java que utiliza essa tecnologia.

primeira fase de avaliação qualitativa (e subjetiva) do mecanismo, foi utilizado o modo transitório dinâmico sem autenticação SSO nos IdPs.

A pesquisa de satisfação foi respondida por quinze (15) profissionais de TI que trabalham em instituições governamentais e por vinte e quatro (24) de profissionais de TI que trabalham em empresas que prestam serviços de TI para o governo, totalizando trinta e nove (39) avaliadores. A aplicação dos questionários serviu para detectar o nível de satisfação dos avaliadores no que tange a utilização do mecanismo agregador de atributos. A seguir, alguns dos resultados obtidos são analisados.

A primeira parte da pesquisa foi dedicada a identificar o conhecimento dos avaliadores em relação a alguns conceitos sobre gestão de identidades (autenticação SSO, IdPs, autenticação federada, SAML, OpenID, OAuth). Com relação a autenticação SSO, 76.9% dos avaliadores responderam ter conhecimento, outros 56.4% sabem o significado de autenticação federada e 66.7% responderam ter conhecimento sobre o que é um IdP. A respeito do SAML, apenas 35.5% dos avaliadores responderam conhecer a tecnologia. Constatou-se que o conhecimento geral dos avaliadores que participaram da pesquisa é limitado, em especial, referente aos conceitos e tecnologias relacionados à gestão de identidades federadas (e.g. SAML).

A pesquisa identificou se as ações no protótipo partiram de uma ação do avaliador (ver Figura 3.a) e se as mensagens de erro (caso tenham ocorrido) ajudaram a resolver o problema (ver Figura 3.b). Os resultados apresentados demonstram que a maioria dos avaliadores respondeu que o protótipo exigiu sua ação. Este resultado é positivo uma vez que o usuário precisa ter consciência de suas ações sobre o sistema e confirma que o protótipo é centrado no usuário. Referente aos erros do sistema, apenas 7.7% dos avaliadores responderam que as mensagens de erros não foram suficientes para contornar o problema. Logo, diante da ocorrência de erros, na maioria das vezes, o sistema contribui para resolução destes.

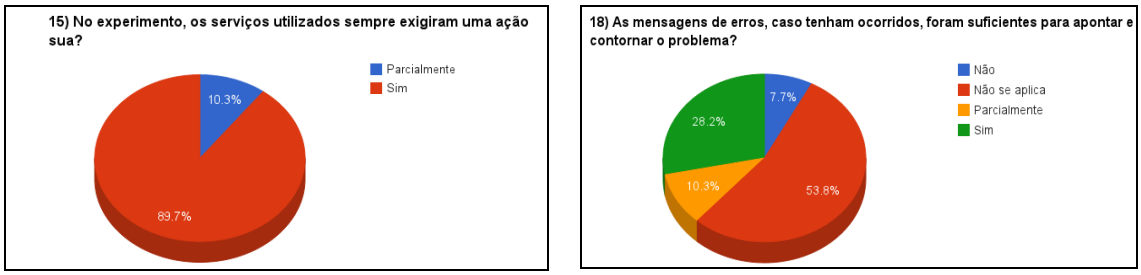


Figura 3 – Resultados da Pesquisa de Satisfação (ações do usuário e msgs de erro)

A Figura 4.a aponta que 76.9% dos avaliadores se sentiram confortáveis durante o uso do mecanismo, o que demonstra uma boa satisfação dos usuários. Alguns dos avaliadores que não se sentiram confortáveis foram os que não conseguiram executar todo o experimento. Outros por acharem o processo “burocrático” diante da necessidade de tantos consentimentos do usuário e de autenticação em todos os IdPs (ausência da autenticação SSO). Outro motivo reportado pelos avaliadores se refere ao uso de

certificados autoassinado[7]. Por fim, um avaliador apontou que o uso do aplicativo executado na máquina do usuário não lhe agrada, que este prefere que todo o processo de agregação seja executado em páginas Web. Não é possível na abordagem mediada pelo cliente esta solicitação do avaliador.

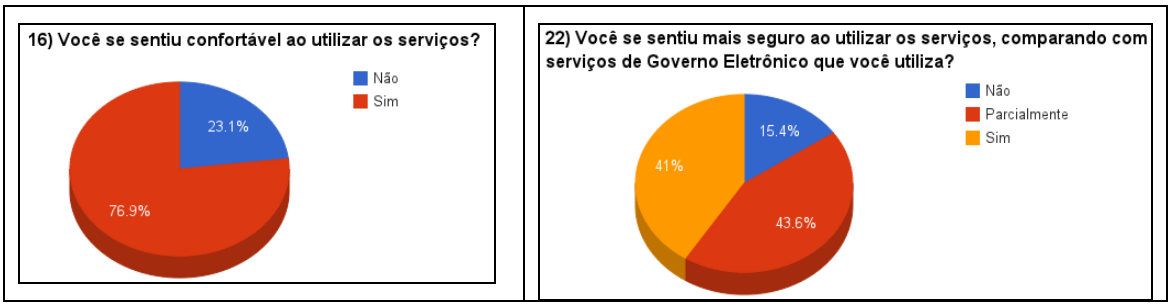


Figura 4 – Resultados da Pesquisa de Satisfação (uso e segurança da aplicação)

Dentre os avaliadores do protótipo do mecanismo, apenas seis responderam não se sentirem seguros ao usar o serviço (Figura 4.b). Dentre estes, quatro não conseguiram completar a execução do experimento e os demais não apontaram seus motivos. É possível que estes consideram os atuais serviços de e.gov seguros ou que o uso de certificados autoassinados prejudicaram este item de avaliação.

Conforme a Figura 5a, a maioria, 97.4% das pessoas que participaram da pesquisa responderam que gostariam de utilizar o mecanismo agregador de atributos em aplicações de governo eletrônico. Verificou-se ainda que apenas um avaliador respondeu que não indicaria a ferramenta. Este avaliador não conseguiu executar todo o experimento e não indicou seus motivos.

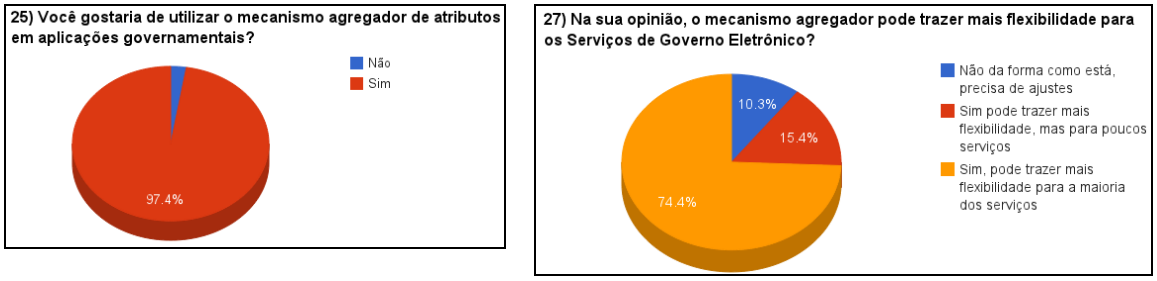


Figura 5: Resultados da Pesquisa (Uso e flexibilidade em serviços de e-Gov)

Em relação à flexibilidade do mecanismo proposto (ver Figura 5b), a maioria dos avaliadores responderam que a solução proposta pode trazer mais flexibilidade para a maioria dos serviços de e-Gov. Os 10,3% que indicam que o sistema necessita de ajustes foram os que não conseguiram executar todos os passos do experimento ou foram os que indicaram a necessidade da autenticação SSO nos IdPs.

Referente à privacidade dos usuários (ver Figura 6), a maioria dos avaliadores respondeu que o mecanismo agregador de atributos pode garantir a privacidade dos

[7] Foi indicado que estes certificados estavam sendo usados por ser tratar de um protótipo experimental e que em um serviço real seriam usados certificados confiáveis, mesmo assim alguns avaliadores criticaram o uso destes certificados.

usuários no processo de coleta de seus atributos. Alguns dos que não avaliam como possível, tiveram problemas na execução do experimento.

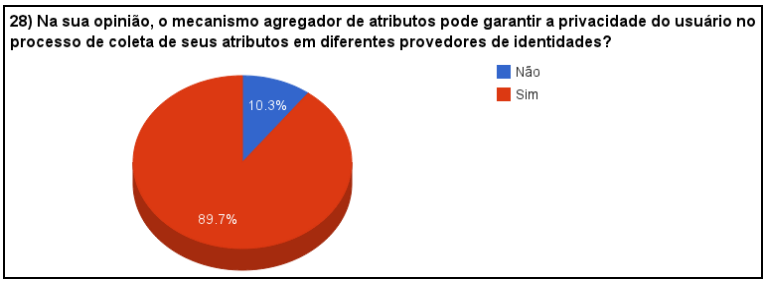


Figura 6: Resultados da Pesquisa de Satisfação (Privacidade dos Usuários)

Em relação à portabilidade do cliente ativo, todos os avaliadores que seguiram as instruções (e.g. não usar *tablets* ou *smartphones*, navegadores web atualizados e máquina virtual Java – JRE 7) executaram com sucesso o experimento.

Segundo os avaliadores existem alguns impactos negativos para os desenvolvedores de aplicação de e.gov na utilização do mecanismo agregador de atributos proposto, tais como: o "Redesenvolvimento" dos mecanismos de autenticação, o que estaria concentrado nos IdPs e não nos SPs como ocorre hoje; a necessidade de aprender uma nova tecnologia, o SAML; a barreira em aceitar que atributos de outros IdPs possam ser utilizados, o que identifica-se uma falta de conhecimento dos benefícios da autenticação SSO federada. Quanto as sugestões do que pode ser melhorado no mecanismo, os avaliadores indicaram: (1) autenticação SSO nos IdPs; (2) política de liberação de atributos para SPs; (3) criação de uma API e documentação para integração e uso do mecanismo agregador de atributos em aplicações de e-Gov.

Na segunda fase de avaliação, foi implementado o modo permanente dinâmico que oferece a autenticação SSO nos IdPs. Na solução, um IdP central é o responsável por autenticar os usuários e prover o *token* de autenticação que é repassado aos demais IdPs. Após a autenticação bem sucedida do usuário, o IdP central, conforme suportando no *simpleSAMLPHP*, indica ao cliente ativo os redirecionamentos que precisam ser efetuados para os IdPs que detêm os atributos requeridos e indicados pelo usuário. De forma semelhante a um navegador, o cliente ativo implementa os redirecionamentos HTTP. Após a implementação, novos testes de software foram executados para verificar o atendimento aos requisitos funcionais e não funcionais de segurança e portabilidade.

6. Conclusão

Diante do desenvolvimento do mecanismo agregador de atributos mediado pelo cliente, da comprovação da aplicabilidade do mecanismo agregador de atributos no cenário de emissão de passaportes, das análises em relação à privacidade, flexibilidade e usabilidade realizadas, é possível afirmar que os objetivos desse trabalho foram atingidos e que a abordagem mediada pelo cliente é viável no cenário de e-Gov.

O mecanismo agregador de atributos apresentado neste artigo inova em relação aos trabalhos relacionados, ao prover uma solução que evita a rastreabilidade dos atributos do usuário, por meio de uma abordagem mediada pelo cliente alinhada à arquitetura ePING. O mecanismo proposto tem como objetivo trazer mais flexibilidade para uma estratégia nacional de gestão de identidades federadas e centrada no usuário.

Por fim, como trabalhos futuros, pretende-se implementar e avaliar o modo permanente dinâmico (definição de uma política de liberação de atributos. Pretende-se ainda implementar o cliente ativo em um cartão inteligente (*tamper-resistant smartcard*) para proteger contra ataques de plataformas de execução maliciosas (ambiente de execução do cliente ativo).

Referências

Baldoni, R. (2012).Federated Identity Management System in e-Government: the Case of Italy, *Electronic Government, an International Journal*, v. 9, no. 1, pp. 64-84.

BRASIL, Comitê Executivo de Governo Eletrônico (2014). ePING – Padrões de Interoperabilidade de Governo Eletrônico. Disponível em: <http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade>. Acesso em: 04 Jul. 2014.

Chadwick, D; Innman, G. Klingenstein, N. (2010). A Conceptual model for Attribute Aggregation, *Future Generation Computer Systems*. vol. 26, no.7, pp.1043-1052.

Chadwick, D.; Inman. G. Siu, K. W . S. Ferdous, M. S (2011). Leveraging Social Networks to Gain Access to Organisational Resources. Proceedings of the 7th ACM workshop on Digital identity management. p. 43-52.

Chadwick, D.; Inman, G., (2013). The Trusted Attribute Aggregation Service (TAAS) - Providing an Attribute Aggregation Layer for Federated Identity Management. Eighth International Conference on Availability, Reliability and Security (ARES), pp.285-290.

Hatakeyma, M.; Shima, S. (2008). Privilege Federation between Different User Profiles for Service Federation, *ACM Conference on Computer and Communications Security*, pp.41-50, New York.

Hoellrigl, T., Kühner, H.; Dinger, J.; Hartenstein, H. (2010). User-Controlled Automated Identity Delegation, *Network and Service Management*, pp. 230-233, Niagara Falls.

Hulsebosch, B.; Wegdam, M.; Zoetekouw, B; Dijk, N.; Poortinga, R. (2011). Virtual collaboration attribute management, *Surf Net: GigaPort3*, vol.1, Sep, 2011.

ITU (2009). Ngn identity management framework. Recommendation Y.2720.

Klingenstein, N. (2007). Attribute Aggregation and Federated Identity, *International Symposium on Applications and the Internet Workshops (SAINTW'07)*, pp. 15-19.

Landau, S.; Gong, H.; Wilton, R. (2009). Achieving Privacy in a Federated Identity Management System, *Financial Cryptography and Data Security*, pp. 51-70, Barbados.

Lee, J. W.; Kim, H.; Hong, J. S.; (2008). An Attribute Aggregation Architecture with Trust-Based Evaluation for Acess Control, *Network Operations and Management Symposium*. pp. 1011-1014, Salvador, 2008.

OECD (2011), National Strategies and Policies for Digital Identity Management in OECD Countries, *OECD Digital Economy Papers*, no. 177, OECD Publishing, 2011.

Vossaert, J.; Lapon, J.; Decker, B. D.; Naessens, V. (2010). User-Centric Identity Management Using Trusted Modules, *European Workshop*, pp.155-170, Atenas.

Monitoração de comportamento de *malware* em sistemas operacionais Windows NT 6.x de 64 bits

Marcus Botacin[1,3], Vitor Afonso[1], Paulo Lício de Geus[1], André Grégio[1,2]

[1] Instituto de Computação (IC)
Universidade Estadual de Campinas (Unicamp)
Campinas – SP – Brasil

[2]Divisão de Segurança de Sistemas de Informação (DSSI)
Centro de Tecnologia da Informação Renato Archer (CTI)
Campinas – SP – Brasil

[3]Bolsista PIBIC-CNPq

{marcus, vitor, paulo}@lasca.ic.unicamp.br, andre.gregio@cti.gov.br

Abstract. *Malware are persistent threats to systems security that are constantly evolving to prevent detection and dynamic analysis techniques. Currently, there is no known dynamic analysis system (publicly available or described in the literature) that supports 64-bits malware (PE+ format). It is difficult to monitor malware for Windows NT 6.x due to new security mechanisms introduced in these systems, making it expensive to build or port an actual analysis system/tool. In this paper, we present the design and implementation of a novel malware dynamic analysis system for Windows 8, as well as the obstacles and challenges we faced. We present the tests and results of the proposed system, evaluated with 2,937 32 and 64-bit malware samples.*

Resumo. *Programas maliciosos (*malware*) são ameaças persistentes à segurança, evoluindo constantemente para evitar a detecção e análise dinâmica. Atualmente, nenhum dos sistemas descritos na literatura ou disponíveis publicamente suportam* malware *de 64 bits (PE+). A monitoração de* malware *em Windows NT 6.x é dificultada devido à introdução de novos mecanismos de segurança, tornando custosa a construção ou portabilidade de um sistema de análise funcional. Neste artigo, apresenta-se o projeto e a implementação de um sistema de análise dinâmica de* malware *em Windows 8, os obstáculos e desafios encontrados e sua avaliação com 2.937 exemplares de 32 e 64 bits.*

1. Introdução

Programas maliciosos têm sido a ameaça mais grave e persistente para a segurança de sistemas interconectados em rede e seus usuários. Esses programas, genericamente chamados de *malware*, subvertem a operação legítima de um sistema computacional de modo a violar sua integridade, confidencialidade ou disponibilidade. Ataques por *malware* são motivados por inúmeros resultados espúrios, tais como evasão de informações, roubo de credenciais, forja de identidade, armazenamento de conteúdo ilícito, lançamento de ataques contra terceiros e ganhos financeiros. Durante um ataque, traços de atividades feitas pelo exemplar de *malware* podem ser observados no sistema operacional. Como exemplo desses traços, pode-se citar arquivos obtidos da Internet, substituição de bibliotecas

ou aplicações do sistema, modificação de chaves do Registro do sistema operacional e alteração em configurações relacionadas com mecanismos de segurança e atualização. A captura desses traços envolve a interceptação das ações realizadas por um programa monitorado no sistema operacional alvo e depende de vários fatores, incluindo o nível no qual a interceptação irá ocorrer (do usuário, do *kernel* ou ainda na camada entre o sistema de virtualização e o sistema base), os mecanismos de proteção existentes no próprio sistema operacional e a capacidade do *malware* analisado em evadir a monitoração.

Os sistemas operacionais Windows ainda são os alvos principais dos criadores de *malware* para computadores pessoais. Embora os sistemas Windows NT com versões acima de 6 (Vista, 7, 8 e 8.1) introduzam diversos mecanismos novos para aumentar a segurança, eles apresentam compatibilidade com aplicações feitas para NT 5, podendo portanto executar programas em 32 e 64 bits. Devido às diferenças entre os sistemas operacionais Windows NT 5.1 (XP) e NT 6.2 (8) ocorrerem em vários níveis (*kernel*, mecanismos de segurança, modo de execução), faz-se necessária a compreensão de como os exemplares de *malware* se comportam durante a execução no Windows 8 e como se dá a interação entre o sistema operacional, o programa malicioso e a ferramenta de monitoração. Mais do que isso, é necessário projetar e implementar uma nova ferramenta de monitoração de comportamento de execução de programas para atuar em sistemas de 64 bits, dado que os analisadores de *malware* publicamente disponíveis não estão prontos para monitorar programas maliciosos de 64 bits.

Neste artigo propõe-se um sistema de análise dinâmica de *malware* baseado em Windows 8. As principais contribuições deste trabalho são: o levantamento e a descrição dos novos mecanismos de segurança presentes em Windows NT 6.x e como estes afetam o desenvolvimento de uma ferramenta para monitoração de ações de programas em execução; o projeto e a implementação de uma arquitetura de análise de *malware* para sistemas e exemplares de 64 bits, com as decisões tomadas, desafios e especificações; a identificação do comportamento suspeito observado nos resultados de execução de mais de 2 mil exemplares de *malware* obtidos nos testes para validação do sistema proposto, bem como a constatação de que *malware* codificado (e compilado) para sistemas Windows XP de 32 bits é capaz de infectar os sistemas novos de 64 bits.

O restante do artigo é dividido da seguinte forma: a Seção 2 trata dos aspectos técnicos distintos introduzidos pela versão NT 6.x dos sistemas operacionais Windows e suas implicações na implantação de uma ferramenta de monitoração de chamadas de sistema e de API nos níveis de *kernel* e de usuário; na Seção 3, são discutidos os principais trabalhos relacionados com sistemas de análise dinâmica de *malware* voltados para obtenção do comportamento, as técnicas utilizadas em sua implementação e as limitações encontradas; na Seção 4, os detalhes do projeto e implementação da arquitetura proposta são expostos, bem como as decisões tomadas para balancear as vantagens e desvantagens das técnicas de monitoração, abrangência de captura e flexibilidade de aplicação (emulação e *bare-metal*); a Seção 5 apresenta os testes realizados na validação do funcionamento do sistema, em especial da ferramenta desenvolvida para monitoração das ações em nível de *kernel*, além de mostrar os resultados da análise de mais de dois mil exemplares coletados recentemente e em atividade. A conclusão do artigo está na Seção 6.

2. Aspectos Técnicos do NT 6.x

A família de sistemas operacionais Windows versão NT 6.x—iniciada com o Windows Vista e da qual o Windows 8 faz parte—traz uma série de diferenças em relação ao Windows XP. A compreensão desses novos mecanismos de segurança e modos de operação é fundamental para se projetar uma ferramenta de monitoração de programas, pois eles afetam diretamente a operação de aplicações no nível do usuário e do *kernel*. Nessa seção, são discutidas as diferenças introduzidas nos Windows modernos e suas implicações para o desenvolvimento de um sistema de análise de *malware*.

2.1. Kernel Patch Protection (KPP)

A proteção de *patch de kernel*, como definida em [Microsoft 2013c], proíbe que *drivers* em nível privilegiado estendam ou substituam serviços do *kernel* por meios não documentados. Tal proibição visa aumentar a segurança do sistema, uma vez que, além dos usos por programas legítimos, muitos *rootkits* utilizam-se desta possibilidade para infectar o sistema. Este mecanismo de proteção de *kernel* encontra-se presente apenas nas versões 64 bits do sistema operacional, dado que para as versões de 32 bits existe uma gama enorme de aplicativos lançados baseados em tais *patches* que viriam a se tornar incompatíveis caso o mecanismo fosse incorporado a todas as edições do sistema. Deve-se notar que a base instalada de aplicativos dependentes destes *patches* para a arquitetura de 64 bits é significativamente menor. O referido mecanismo visa impedir:

- Modificações nas tabelas de serviços do sistema, por exemplo, conectar-se à tabela `KeServiceDescriptor`;
- Modificações na IDT (*Interrupt Descriptor Table*);
- Modificações na GDT (*Global Descriptor Table*);
- Uso de pilhas de *kernel* que não sejam alocadas por este;
- Aplicar *patches* em qualquer parte do *kernel* (somente AMD64).

Uma implicação que KPP traz para a monitoração dinâmica de *malware* é na implementação de técnicas de *hooking*, as quais consistem na alteração das funções originais do sistema por funções de interceptação, responsáveis por coletar os dados monitorados e dar continuidade às ações originalmente pretendidas. Essa técnica, quando aplicada no nível do *kernel*, se mostra altamente efetiva, uma vez que captura dados em um nível de execução privilegiado, está devidamente isolada do espaço de execução do *malware* e afeta todo o sistema, não necessitando ser aplicada a cada processo individualmente.

A implementação de *hooks* de *kernel* é realizada através da substituição dos endereços das funções diretamente nas tabelas exportadas pelo *kernel*. Tal substituição é dificultada no Windows 8 devido ao mecanismo de KPP, dado que quando se tenta implementar esse tipo de *hooking* em 64 bits, verifica-se que as tabelas de funções não são mais exportadas pelo *kernel*. Além disso, sua localização é imprevisível, uma vez que há um mecanismo de aleatorização de espaço de memória. Desta forma, deve-se identificar novos meios de interceptação no nível do *kernel* que não sejam baseados em *hooking*.

Uma outra forma de interceptar APIs do Windows é através do uso de *Detours*, uma biblioteca provida pela Microsoft. Sua utilização permite realizar modificações dinâmicas (durante a execução do programa) no início da função que se deseja interceptar através da inserção de uma instrução *assembly* de pulo incondicional (`JMP`). O uso de *Detours*, no entanto, passou a ser condicionado a aquisição de licenças [Microsoft 2013b].

2.2. Assinatura de *Drivers*

Com o objetivo de aumentar a segurança no sistema de modo a evitar que componentes arbitrários sejam carregados no *kernel*, a Microsoft passou a exigir que os *drivers* sejam assinados digitalmente. Tal política de desenvolvimento impede, a princípio, a utilização de um *driver* como mecanismo de captura se este não for assinado digitalmente, o que não é compatível com os requisitos de uma ferramenta de monitoração. No entanto, esta proteção pode ser desligada no ambiente de análise, pois os exemplares de *malware* em geral atuam em espaço de usuário e não apresentam a característica de carregar *drivers* no sistema. Entretanto, o desligamento desse sistema facilita a atuação de *rootkits*, sendo que estes, assim como ocorre em todos os sistemas tradicionais de análise dinâmica de *malware*, não tem sua execução monitorada pela ferramenta proposta.

2.3. Sessões

De modo a tentar isolar diferentes famílias de aplicações (aplicações gráficas, serviços de sistema, serviços remotos), o sistema operacional Windows passou a implementar o conceito de sessões, no qual os dados e privilégios de execução de cada uma dessas famílias ficam restritos às mesmas. Essa mudança trouxe impactos significativos, como o impedimento do lançamento de *threads* remotas, comumente usadas para injeções de *Dynamic Link Library* (DLL) [Microsoft 2013a]. Dessa forma, minimiza-se a chance de sucesso dos ataques por injeção de DLL aos *browsers*, ao mesmo tempo em que se dificulta o monitoramento de atividades suspeitas através de *DLL hooking*.

2.4. Mudanças na API

As diferenças arquiteturais promovidas do Windows XP para o Windows Vista, e consequentemente para o Windows 8, trouxeram mudanças nas interfaces de programação. Embora as interfaces antigas, em geral, ainda funcionem, deve-se realizar a migração para as novas interfaces de forma que se possa utilizar todos os recursos disponíveis. Uma mudança significativa da API pode ser vista em [Microsoft 2014a] e [Microsoft 2014b].

As interfaces providas pelo próprio sistema operacional provêem a possibilidade de uso na interceptação das ações realizadas e, consequentemente, na análise dinâmica de *malware*. Ao se projetar uma ferramenta de análise, deve-se atentar para as interfaces utilizadas pois estas podem sofrer modificações nas atualizações do sistema, dificultando a portabilidade do mecanismo de monitoração.

Interfaces como *callbacks* e *filters* estão disponíveis a partir do *kernel* por meio do uso de *drivers*. Por serem providos pelo sistema operacional, *callbacks* e *filters* apresentam interfaces e estruturas bem definidas, além de serem bem documentados, o que facilita o desenvolvimento. Adicionalmente, não requerem qualquer mudança em estruturas internas do *kernel* ou de bibliotecas dinâmicas, possuindo suporte nativo no sistema. A desvantagem no uso dessas técnicas é que a captura de informações é limitada às funcionalidades providas pelas interfaces utilizadas.

3. Trabalhos Relacionados

Há diversas ferramentas para a monitoração do comportamento de execução de programas maliciosos, as quais aplicam diferentes técnicas para interceptar as ações efetuadas. Nesta

seção, serão avaliados alguns sistemas de análise dinâmica de *malware* e suas características principais, ressaltando que todos eles suportam apenas executáveis de 32 bits.

Anubis [Bayer et al. 2006] é um sistema de análise dinâmica que se utiliza da técnica de *Virtual Machine Introspection* (VMI) aplicada ao emulador Qemu [Bellard 2005]. Com essa técnica, cria-se uma camada entre o sistema de análise (*guest*) e o ambiente-base (*host*) para processamento e controle, possibilitando que a interceptação das ações executadas pelo *malware* dentro do ambiente de análise seja feita sem que haja qualquer interferência neste. Isso torna possível que um dado *malware* seja analisado sem qualquer tipo de modificação "visível", seja no sistema *guest*, seja no próprio *malware*. O sistema operacional utilizado pelo Anubis no ambiente de análise é o Windows XP SP3, do qual são capturados diversos tipos de atividade durante a execução do *malware* (sistema de arquivos, processos, Registro, objetos de sincronização e tráfego de rede). Ao final da execução, é produzido um relatório técnico sobre a análise do exemplar submetido.

No CWSandbox [Willems et al. 2007], a captura das informações é realizada por uma DLL, a qual precisa ser injetada no processo do *malware*. Quando a DLL é carregada, as principais funções utilizadas para fazer a interface entre o programa e o sistema de análise (por exemplo, modificações em arquivos) têm seu início modificado. Desta forma, um desvio incondicional é executado assim que uma dada função é chamada. Para iniciar o procedimento de análise, existe um componente dentro do ambiente de monitoração—`c`wsandbox.exe—cujas funções são criar o processo do *malware* em estado suspenso, injetar a DLL e retomar a execução do processo em questão. Além disso, este componente é informado caso o *malware* inicialize ou modifique algum processo, para que a DLL de monitoração seja injetada neles também. Finalizada a análise, é gerado um relatório em diversos formatos (HTML, XML e texto), o qual contém as ações realizadas pelo *malware* no sistema operacional monitorado.

Cuckoo [Guarnieri 2013] utiliza uma técnica conhecida como *inline hooking* para interceptar as chamadas de sistema executadas pelo programa a ser monitorado. Para implementar *inline hooking*, Cuckoo precisa carregar uma DLL no processo que se deseja monitorar. O *hooking* é implementado de forma específica para cada função interceptada, diferentemente de um simples salto incondicional no início da função. Isto dificulta métodos triviais de detecção, buscando evitar assim que a análise não seja bem sucedida. Atualmente, existem dois métodos de monitoração implementados, os quais são escolhidos de forma aleatória no momento que o *inline hooking* é instalado. A DLL do Cuckoo é carregada por um *script* em linguagem Python que permanece em execução no ambiente de análise durante todo o processo. Além da tarefa de carregamento, tal *script* é notificado caso o exemplar de *malware* monitorado modifique ou crie um novo processo durante a análise, indicando que a DLL de monitoração deve ser carregada nele. O Cuckoo tem seu código disponível para utilização (`http://www.cuckoosandbox.org`), a qual requer a preparação do ambiente com uma instalação local para a realização das análises.

BehEMOT [Filho et al. 2010] é uma ferramenta de análise dinâmica de *malware* para Windows XP que pode monitorar as principais interações entre um exemplar de *malware* (e seus processos-filhos) e o sistema operacional alvo, como operações em arquivos, chaves do Registro, processos e objetos de sincronização (*mutex*). Além da análise comportamental do *malware* no sistema operacional, a ferramenta realiza a captura de tráfego de rede, provendo informações sobre a interação do exemplar monitorado com

o ambiente externo, como servidores comprometidos hospedando outros objetos maliciosos ou armazenando informações sensíveis. Visando contornar as possíveis técnicas de anti-análise utilizadas por alguns *malware* para detectar máquinas virtuais, a ferramenta baseia-se em uma arquitetura mista de ambientes emulado e real, de modo que os exemplares que não podem ser analisados no ambiente emulado são encaminhados para o ambiente em *bare metal*. Esta característica de flexibilidade entre os ambientes é alcançada devido ao componente de monitoração ter sido implementado como um *driver* de *kernel* que aplica a técnica de *SSDT hooking*, que permite capturar uma vasta gama de ações em nível privilegiado por meio de chamadas de sistema nativas.

Capture-BAT [Seifert et al. 2007] é uma ferramenta de análise de *malware* baseada em *drivers* de *kernel*, objetivando, sobretudo, a portabilidade. O Capture-BAT executa em Windows XP SP2 e monitora as operações de "READ" e "WRITE" através de um *filesystem filter*, e a criação/término de processos e as operações em chaves de Registro através de *kernel callbacks*. Deve-se destacar que, diferentemente das ferramentas descritas anteriormente, não há propagação das ações sobre processos em relação aos demais componentes, isto é, a captura das informações é feita em modo "*system-wide*". Além disso, tanto no caso dos processos, quanto no do Registro, armazena-se apenas algumas informações dentre as possíveis de serem coletadas, como o *timestamp* (em ambos os casos), o *path* do processo (em sua *callback*) e o *path* da chave (no monitor de registro). Portanto, informações mais detalhadas, como o valor de uma chave escrita no Registro e os parâmetros passados para um dado processo, ficam a cargo de outros programas.

3.1. Considerações sobre as ferramentas avaliadas

Com base nos trabalhos relacionados e suas características de operação, as seguintes considerações foram elencadas:

- Anubis fornece um bom referencial para obter abrangência de monitoração e exibição dos resultados. No entanto, verifica-se que apesar do sistema operacional "*guest*" não sofrer alterações, o mecanismo de análise é dependente do Qemu. Isso faz com que o Anubis não funcione com outro tipo de tecnologia de virtualização nem possa ter seu mecanismo de análise instalado em máquinas reais (*bare metal*), o que é uma característica desejável ao sistema proposto neste artigo;
- CWSandbox, por sua vez, não seria funcional, nem poderia ser implementado em sistemas Windows 8 da mesma forma que está atualmente, dado que se baseia em injeção de DLLs. Isto faz com que ele sofra das limitações expostas na Seção 2.3. Além disso, há a necessidade de uma nova injeção de DLL para cada processo criado, tornando o mecanismo de monitoração muito custoso;
- Cuckoo Sandbox, além das restrições contra injeção de DLLs, o fato de o código ser aberto e disponível facilita que os atacantes criem mecanismos de anti-análise e impeçam a monitoração adequada de seus exemplares de *malware*.
- BehEMOT, por ser baseado em um *driver*, é flexível o suficiente para atuar em máquinas virtuais e reais, além de executar em um nível mais privilegiado que a maioria dos exemplares de *malware*. No entanto, dado que o mecanismo de interceptação de chamadas de sistema é implementado por meio da técnica de *SSDT hooking*, seu funcionamento em Windows 8 não é possível (Seção 2.1).
- Capture-BAT, por ser implementado sob a forma de *filter drivers*, permite flexibilidade de atuação, pois exige apenas a instalação do *driver* no sistema *guest* ou

real (ambiente monitorado). Logo, pode operar sobre diversos tipos de sistemas, como *bare metal*, VirtualBox, KVM, VMWARE, entre outros. Além disso, devido ao fato da implementação por *filter drivers* ser um recurso nativo do sistema operacional, esta não sofre das limitações impostas às DLLs, embora exija sua assinatura, como descrito na Seção 2.2.

4. Projeto do Sistema Proposto

Nesta seção, as decisões de projeto são expostas, bem como detalhes da implementação realizada e da arquitetura proposta para o sistema de análise.

4.1. Decisões de Projeto e Implementação

Para o desenvolvimento da ferramenta, deve-se considerar as limitações das ferramentas/sistemas mencionados na Seção 3.1 e relacionadas a *hooks* em *kernel* descritas na Seção 2.1. Rossow et al. [Rossow et al. 2012] estabelece que o método de monitoração deve atuar em um nível mais privilegiado do que o objeto sob análise para minimizar o risco de detecção, sabotagem ou subversão da ferramenta por parte do *malware*, que pode inclusive verificar sua integridade em memória a fim de identificar mecanismos de monitoração (ex.: injeção de DLL feita pela Cuckoo Sandbox). Dessa forma, decidiu-se implementar a ferramenta de análise através de um *driver* de *kernel*, consistindo na mesma abordagem usada na implementação da ferramenta Capture-BAT, porém capturando o tráfego de rede externamente ao ambiente de execução do *malware*. Embora Capture-BAT faça uso apenas da técnica de *filter driver*, deve-se lembrar que sua codificação baseia-se em Windows XP de 32 bits. Portanto, a implementação de uma solução utilizando a mesma técnica em Windows 8 de 64 bits requer a adequação dos tipos de dados (devido ao tamanho da palavra) e dos parâmetros de chamadas de *callback* (Seção 2.4).

Cabe ressaltar que a monitoração feita por Capture-BAT é insuficiente para a avaliação adequada do comportamento exibido por *malware* sob análise, uma vez que não são obtidas certas informações desejáveis, como os valores escritos nas chaves de Registro criadas ou modificadas. Além disso, Capture-BAT é incapaz de monitorar apenas um determinado processo escolhido, interceptando as modificações feitas no sistema operacional como um todo, inclusive as feitas por programas legítimos do próprio sistema que estão em execução em *background*. Na ferramenta de monitoração proposta neste artigo, foi implementado suporte à captura dos valores escritos nos campos alterados e foram desenvolvidos métodos para interceptar apenas o processo submetido para análise e seus processos-filhos ou aqueles cujos processos monitorados interagiram. Logo, concentrou-se esforços na produção de uma solução funcional para Windows 8 com a versatilidade apresentada por Capture-BAT, porém com suas principais limitações resolvidas.

A ferramenta desenvolvida é um *driver* de *kernel* que aplica duas técnicas distintas para interceptação das ações realizadas pelo programa monitorado: ações de **Registro** e de **processos** são obtidas por *callbacks* (Figura 1) e ações do **sistema de arquivos** por um *filesystem filter*. As informações armazenadas em *log* durante a execucão de um dado exemplar de *malware* são: registro temporal (*timestamp*); processo originário da ação (PID e nome); ação realizada (escrita de arquivo, leitura de arquivo, remoção de arquivo, criação de processo, término de processo, definição de chave do Registro ou remoção de chave do Registro); parâmetro/valor da ação; objeto-alvo da ação. Adicionalmente, a **remoção** de arquivos armazena o arquivo a ser removido para análise posterior.

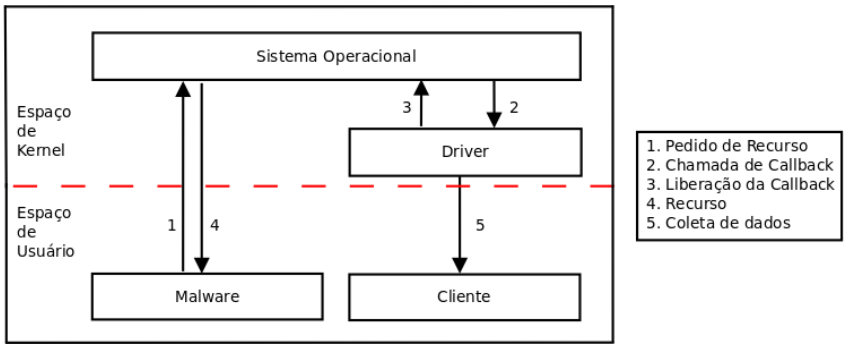


Figura 1. Passos realizados na interceptação por técnica de *callback*.

4.2. Arquitetura do Sistema

A partir do *driver* desenvolvido, projetou-se um sistema automatizado para análise dinâmica com diversos outros componentes auxiliares, como a captura de tráfego de rede e o controle da sua saída. O *driver* provê a comunicação com o nível de usuário de modo que diferentes programas clientes podem obter as informações capturadas através do uso de *I/O Request Packets* (IRP). Um dos componentes implementados é o "CLIENTE", um programa interno ao ambiente da análise que se comunica constantemente com o *driver*, responsável por obter os dados capturados durante a monitoração do *malware*. Este programa também permite alterar configurações básicas do funcionamento do *driver*, como registrar os subsistemas a serem monitorados e ligar/desligar o modo de *debugging*.

Outro componente importante do sistema proposto é o "CONTROLADOR DA ANÁLISE", que consiste de uma aplicação interna ao ambiente de análise, a qual contém suporte a requisições de rede, utilizando-se de um *socket* TCP para obtenção e execução do *malware* via "CONTROLADOR EXTERNO", envio de resultados e recebimento de comandos de controle. Esse componente também é responsável por agrupar os *logs* gerados, o tráfego de rede capturado e os arquivos removidos em um único arquivo a ser transmitido ao ambiente externo.

No âmbito da rede, o tráfego gerado durante a execução do *malware* é capturado externamente ao ambiente da análise e armazenado em formato `pcap` via `tcpdump`. O tráfego de rede capturado durante a execução do exemplar de *malware* é separado do tráfego originado por aplicações do sistema operacional através de filtros elaborados com base na execução de um sistema não contaminado. As boas práticas para a análise de *malware* englobam a execução em ambientes totalmente controlados, de forma a evitar ataques contra terceiros e contaminações. Para tanto, todo o tráfego de saída passa por um *firewall* (`IPTables`) que permite conexões HTTP e HTTPS para que o *malware* possa fazer *download* e verificar a conectividade. outros protocolos. O tráfego relacionado aos demais protocolos de aplicação é redirecionado para um *honeypot* (`honeyd`) para fins de registro dos demais protocolos utilizados. A integração dos componentes em uma arquitetura baseada em máquinas virtuais, para aumentar a escalabilidade, resulta no sistema proposto ilustrado na Figura 2. Cabe ressaltar que o sistema pode atuar em máquinas *bare-metal* sem a necessidade de alteração de código, bastando que se configure o ambiente e seus componentes adequadamente.

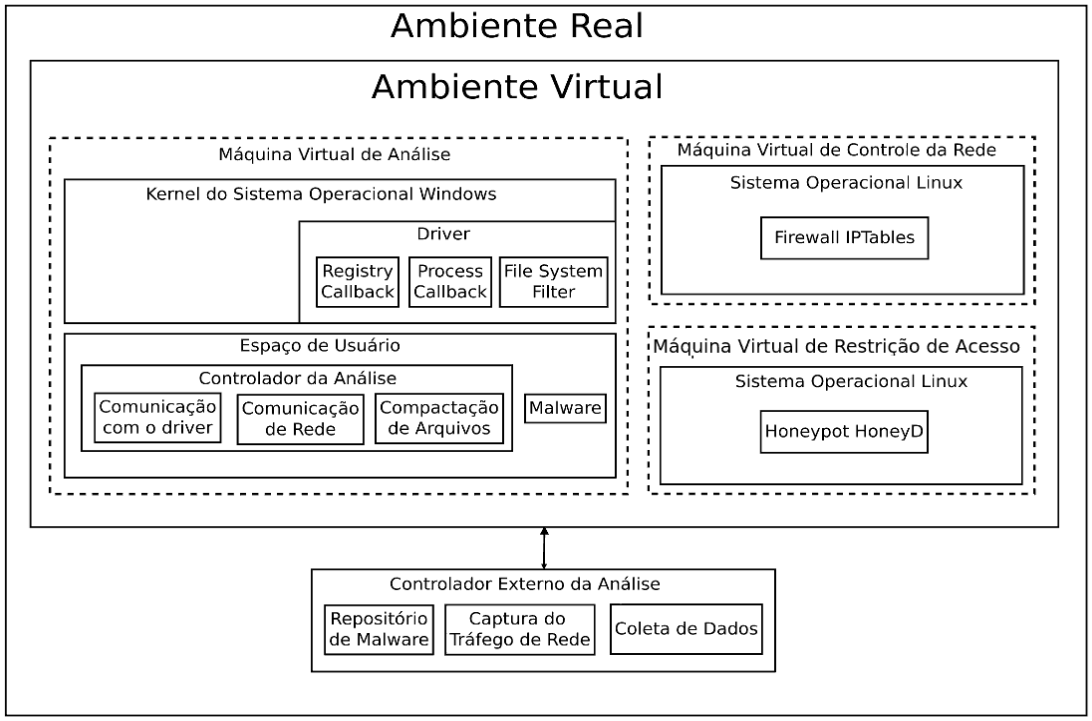


Figura 2. Arquitetura do sistema de análise de *malware* (Windows 8, 64 bits).

5. Testes e Resultados

Nesta seção são apresentados os testes realizados para validar o correto funcionamento do sistema proposto, bem como resultados obtidos da análise. No período entre 01/01/2014 e 21/05/2014 foram coletados 2.937 exemplares de *malware* únicos (com base no *hash* MD5) provenientes de *honeypots*, *phishing* e *downloads* de *links* contaminados. O ambiente de análise definido para os testes consiste de uma máquina virtual com sistema operacional Windows 8 de 64 bits sem qualquer mecanismo de segurança habilitado, de modo a evitar eventuais interferências com o *malware* a ser executado ou com a ferramenta de monitoração desenvolvida. O mecanismo de virtualização utilizado é o Qemu-KVM em operação sobre um *host* Linux Ubuntu 12.04 Server.

5.1. Validação

A fim de verificar se a monitoração das ações efetuadas sobre os subsistemas de arquivos, Registro e processos é feita adequadamente, foram executados alguns exemplares da coleção obtida para este artigo. Após a execução desses exemplares no sistema proposto, trechos foram escolhidos para ilustrar que as operações definidas são monitoradas com sucesso.

A Listagem 1 mostra o exemplar "7G6C5n.exe" definindo o valor "C:\7G6C5n.exe" na chave "...\Windows\CurrentVersion\Run\SoftBrue" através da ação `SetValueKey`. Com isso, o exemplar é executado durante a inicialização do sistema operacional, fazendo com que o *malware* sobreviva ao desligamento da máquina ou a um eventual *reboot*.

Listagem 1. Monitoração de ação de escrita em chave do Registro.

```
1 7/4/2014 -13:3:48.793|SetValueKey|2032|C:\7G6C5n.exe|\REGISTRY\
    USER\S-1-5-21-3760592576-961097288-785014024-1001\Software\
    Microsoft\Windows\CurrentVersion\Run|SoftBrue|"C:\7G6C5n.exe"
```

A Listagem 2 mostra o exemplar "visualizar.exe" escrevendo dados (`WriteOperation`) no arquivo "dll.exe", um programa do sistema. Esse tipo de ação, isto é, a modificação de um arquivo existente, pode causar a inclusão de funcionalidades maliciosas em programas legítimos.

Listagem 2. Captura de ação de escrita no sistema de arquivos.

```
1 7/4/2014 -13:3:48.76|WriteOperation|3028|C:\visualizar.exe|C:\
    Windows\SysWOW64\dll.exe|
```

Já a Listagem 3 mostra o *malware* "deposito.exe" efetuando a remoção do arquivo "rr.txt". Tal arquivo pode ter sido gerado anteriormente pelo exemplar para armazenar alguma informação obtida e a ação `DeleteOperation` indica a remoção de evidências da infecção do sistema-alvo.

Listagem 3. Ação de remoção de arquivo no sistema-alvo.

```
1 7/4/2014 -13:5:1.895|DeleteOperation|2032|C:\deposito.exe|C:\
    ProgramData\rr.txt|
```

A Listagem 4 mostra o programa "visualizar.exe" chamando um programa do sistema modificado anteriormente, como ilustrado na Listagem 2, por meio da criação do processo "dll.exe" (ação `CreateProcess`).

Listagem 4. Processo monitorado devido a interação com malware.

```
1 7/4/2014 -13:3:48.294|CreateProcess|3028|C:\Monitor\Malware\
    visualizar.exe|2440|C:\Windows\SysWOW64\dll.exe
```

A Listagem 5 mostra uma sessão de rede na qual o *malware* acessa a porta 80 (HTTP) de um endereço IP comprometido e efetua a requisição `GET`.

Listagem 5. Exemplo de tráfego de rede capturado durante análise.

```
1 2014-05-14 20:02:40.963113 10.10.100.101 XX.YY.ZZ.121
    HTTP 290 GET /.swim01/control.php?ia&mi=00B5AB4E-47098
    BC3 HTTP/1.1
```

Logo, observa-se que todos os tipos de ações que deveriam ser monitoradas são armazenadas em *logs*, validando o funcionamento do sistema e possibilitando o teste geral com todos os exemplares de *malware* da coleção.

5.2. Resultados com *Malware*

Após verificar que que os resultados produzidos pelo sistema estão corretos, inclusive utilizando programas especialmente feitos para testar suas funcionalidades de monitoração, os 2.937 exemplares coletados foram submetidos para análise dinâmica no sistema proposto. A variedade dos tipos de arquivos nos quais os exemplares encontram-se contidos

é ilustrada na Figura 3. PE (*Portable Executable*) é um formato de arquivo utilizado em sistemas Windows para arquivos executáveis e bibliotecas, entre outros tipos de arquivo, tendo uma versão de 32 bits (PE32) e outra de 64 bits (PE+). DLLs (*Dynamic-Link Libraries*) são bibliotecas compartilhadas no formato PE. Arquivos CPL são um tipo especial de DLL que exportam a função *CPlApplet*. Este tipo de arquivo é usado pelos *applets* do painel de controle do sistema Windows e pode ser executado diretamente por usuários via clique duplo. Mono é uma plataforma de *software* que permite o desenvolvimento de aplicações multi-plataforma. Além disso, ele é uma implementação de código aberto do *framework* .NET da Microsoft.

Todos os 2.937 exemplares de *malware* foram submetidos ao VirusTotal (`http://www.virustotal.com`), um serviço *online* que analisa arquivos com 53 antivírus disponíveis no mercado e fornece as assinaturas de identificação encontradas por cada um deles (rótulos de detecção). Deste total, 2.520 exemplares foram detectados por pelo menos um antivírus na época da submissão, o que alarma pela quantidade de exemplares que podem ter sido usados para infecção de usuários nos primeiros dias de disseminação ($\approx$ 15%). A Figura 4 mostra os 10 rótulos de detecção mais atribuídos ao total de exemplares da coleção.

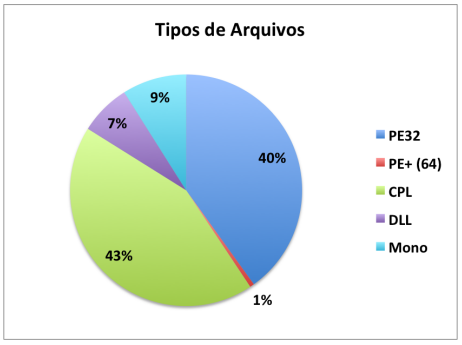


Figura 3. Distribuição de amostras por tipo de arquivo.

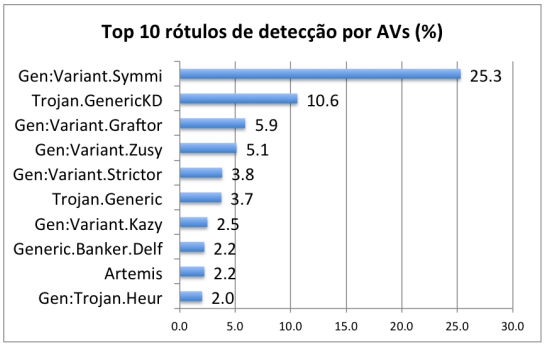


Figura 4. Rótulos de detecção observados (porcentagem).

5.2.1. Comportamentos suspeitos observados no sistema operacional

Na Tabela 1, são mostradas as atividades monitoradas durante a execução dos exemplares de *malware* no sistema proposto neste artigo, bem como quantos dos 2.937 analisados as apresentaram. Deste total, 55 programas não produziram resultados, o que pode ocorrer por diversas razões, tais como não ter acesso a algum componente necessário para a continuidade da execução, o arquivo estar corrompido ou realizar alguma ação não permitida no Windows 8, a identificação da execução em ambiente emulado, entre outras.

A análise mais aprofundada dessas atividades revelou os seguintes comportamentos suspeitos típicos de uma infecção:

- Finalização de mecanismos antívirus instalados no sistema operacional;
- Desligamento do *firewall* nativo do Windows;
- Criação de novos binários no sistema, seja por *download* ou por *dropping*;
- Desligamento do mecanismo de atualização automática do Windows;

Tabela 1. Atividades monitoradas e quantidade de exemplares que as exibiram.

Atividade	Qtde.
Escrita no Registro	1073
Remoção de chave(s) do Registro	772
Criação de processo(s)	602
Término de processos	1337
Escrita em arquivo(s)	1028
Leitura de arquivo(s)	1694
Remoção de arquivo(s)	551

- Tentativa de persistência (sobrevivência a desligamentos e reinicializações);
- Injeção de *Browser Helper Objects* no Internet Explorer;
- Modificação no arquivo `hosts.txt` do sistema operacional;
- Sobrescrita de um arquivo (programa ou biblioteca) já presente no sistema;
- Remoção de seu próprio programa ou de outros artefatos.

5.2.2. Comportamentos suspeitos observados no tráfego de rede

A análise do tráfego de rede capturado durante a execução dos exemplares traz uma perspectiva adicional para o entendimento da atuação do *malware* no sistema infectado. A Tabela 2 mostra os protocolos e portas que mais foram utilizadas pelos exemplares analisados. Nota-se que quase metade deles fazem uso do protocolo HTTP para buscar novos componentes ou enviar dados para o atacante, uma vez que a porta 80 geralmente não é bloqueada na saída. Também é interessante ressaltar as atividades das outras portas, obtidas após análise manual do tráfego capturado: a porta 9000 foi utilizada com destino de comunicação similar à de *bots*, recebendo dados com um formato parecido com JSON; a porta 2869 foi utilizada para troca de tráfego HTTP; nenhuma tentativa de comunicação com a porta 720 teve sucesso no fechamento do *3-way handshake*; a porta 82 foi utilizada tanto para recebimento de tráfego HTTP como para tráfego aparentemente codificado; a porta 8181 foi utilizada no recebimento de informações evadidas do sistema-alvo, sem prover resposta do lado do servidor.

Tabela 2. Top 10 Protocolos/portas mais utilizados por *malware* (% do total de exemplares) observados no tráfego de rede capturado durante a análise.

Proto	HTTP	HTTPS	MS-SQL	-	SMTP	-	MySQL	-	-	-
Porta	80	443	1433	8181	587	82	3306	720	2869	9000
Qtde.	44,4%	6,5%	2,6%	1,0%	0,8%	0,7%	0,5%	0,3%	0,3%	0,2%

A análise mais aprofundada do tráfego de todos os exemplares em busca de comportamentos que indicam a presença de códigos maliciosos produziu os resultados apresentados na Tabela 3. Tais comportamentos incluem: *Download* desconhecido, isto é, binários executáveis não identificados por mecanismos antivírus; *E-mail/Spam*, que consiste do envio de informações por *e-mail* ou tentativa de envio de *spam*; *Banker*, que indica *malware* que tenta evadir credenciais do usuário (agência, conta, tabela de senhas, senha

do *Internet Banking*); comunicação IRC, na qual são identificados comandos típicos de protocolos de *Instant Relay Chat*; dados do sistema (nome, usuário, versão) evadidos via rede; obtenção de PAC (*Proxy Auto Configuration files*), arquivos carregados no *browser* que modificam a navegação; portas de IRC, que indica que uma porta comumente associada a este tipo de protocolo foi acessada.

Tabela 3. Comportamentos suspeitos observados no tráfego de rede.

Comportamento	Qtde. de *malware*
Download desconhecido	154
E-mail/Spam	25
Banker	22
Comunicação IRC	4
Dados do sistema	3
Obtenção de PAC	1
Portas de IRC	1

Discussão. O sistema proposto é o único de que se tem notícia que é tanto capaz de executar arquivos no formato PE+ (64 bits) quanto de prover um ambiente de 64 bits (Windows 8) para análise de *malware*. Exemplares de 64 bits foram submetidos para os sistemas de análise dinâmica *Anubis* (`http://anubis.iseclab.org`), *Cuckoo*, *ThreatExpert* (`http://www.threatexpert.com`), Camas Comodo (`http://camas.comodo.com`) e CWSandbox (`http://www.threattracksecurity.com/resources/sandbox-malware-analysis.aspx`) em suas versões disponíveis publicamente. Destes, nenhum foi capaz de realizar a análise, seja por não suportar explicitamente o tipo de arquivo ou por não retornar resposta, indicando um *crash* no sistema. Um ponto interessante sobre os exemplares analisados diz respeito aos rótulos de detecção providos pelos antivírus: a maioria deles baseia-se em heurísticas genéricas que, embora permitam que o usuário seja alertado sobre um programa malicioso, não provêem informações sobre o tipo de dano causado. Um sistema de análise dinâmica como o proposto neste artigo complementa uma ferramenta antivírus, provendo informações do comportamento, além de permitir a identificação de programas suspeitos quando ainda não há assinaturas ou heurísiticas codificadas. A constatação mais grave acerca dos resultados obtidos é que, mesmo com os mecanismos de segurança propostos a partir do NT 6, a retrocompatibilidade faz com que exemplares de 32 bits codificados e compilados para Windows XP infectem também os Windows 8 caso o atacante subverta o sistema operacional e desabilite tais mecanismos.

6. Conclusão

Neste artigo, introduziu-se o projeto de arquitetura e a implementação de um sistema de análise dinâmica de *malware* de 64 bits baseado em Windows 8, com suas características, desafios e decisões tomadas. O funcionamento do sistema, único do tipo do qual se tem notícia, foi avaliado por meio da execução de 2.937 exemplares de *malware*, cujos resultados mostraram a utilidade da monitoração das ações no nível da rede e do *kernel* do sistema operacional para a identificação de comportamentos suspeitos. Os resultados obtidos permitem uma maior compreensão da atuação de *malware*, possibilitando a criação de heurísticas de detecção, procedimentos de remediação e tomada de contra-medidas

para resposta a incidentes. Os trabalhos futuros incluem a integração do ambiente *bare-metal* ao ambiente emulado a fim de se monitorar *malware* que possua mecanismo anti-análise, a implementação de técnicas para monitoração de outros subsistemas (como o de gerenciamento de memória) e o estudo e desenvolvimento de mecanismos de proteção para a ferramenta de monitoração, visando evitar a detecção por parte de *malware* mais complexo e consequente evasão da análise.

Referências

Bayer, U., Kruegel, C., and Kirda, E. (2006). Ttanalyze: A tool for analyzing malware. In *15th European Institute for Computer Antivirus Research (EICAR 2006) Annual Conference*.

Bellard, F. (2005). Qemu, a fast and portable dynamic translator. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '05, pages 41–41, Berkeley, CA, USA. USENIX Association.

Filho, D. S. F., Grégio, A. R. A., Afonso, V. M., Santos, R. D. C., Jino, M., and de Geus, P. L. (2010). Análise comportamental de código malicioso através da monitoração de chamadas de sistema e tráfego de rede. *Anais do X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*.

Guarnieri, C. (2013). Cuckoo sandbox. `http://www.cuckoosandbox.org/`. Acesso em junho/2014.

Microsoft (2013a). CreateRemoteThread. `http://msdn.microsoft.com/en-us/library/windows/desktop/ms682437(v=vs.85).aspx`. Acesso em junho/2014.

Microsoft (2013b). Detours. `http://research.microsoft.com/en-us/projects/detours/`. Acesso em junho/2014.

Microsoft (2013c). Kernel patch protection for x64-based operating systems. `http://technet.microsoft.com/pt-br/library/cc759759(v=ws.10).aspx`. Acesso em junho/2014.

Microsoft (2014a). CmRegisterCallback. `http://msdn.microsoft.com/en-us/library/windows/hardware/ff541918(v=vs.85).aspx`. Acesso em junho/2014.

Microsoft (2014b). CmRegisterCallbackEx. `http://msdn.microsoft.com/en-us/library/windows/hardware/ff541921(v=vs.85).aspx`. Acesso em junho/2014.

Rossow, C., Dietrich, C. J., Kreibich, C., Grier, C., Paxson, V., Pohlmann, N., Bos, H., and van Steen, M. (2012). Prudent Practices for Designing Malware Experiments: Status Quo and Outlook . In *Proceedings of the 33rd IEEE Symposium on Security and Privacy (S&P)* , San Francisco, CA.

Seifert, C., Steenson, R., Welch, I., Komisarczuk, P., and Endicott-Popovsky, B. (2007). Capture - a behavioral analysis tool for applications and documents. *Digital Investigation*, 4S:S23–S30.

Willems, C., Holz, T., and Freiling, F. (2007). Toward automated dynamic malware analysis using cwsandbox. *IEEE Security & Privacy*, 5:32–39.

Prevenção de Ataques em Sistemas Distribuídos via Análise de Intervalos

Vitor Mendes Paisante, Luiz Felipe Zafra Saggioro, Raphael Ernani Rodrigues, Leonardo Barbosa Oliveira, Fernando Magno Quintão Pereira

[1]Departamento de Ciência da Computação – UFMG
Av. Antônio Carlos, 6627 – 31.270-010 – Belo Horizonte – MG – Brazil

{paisante,luizfzsaggioro,raphael,leob,fernando}@dcc.ufmg.br

Abstract. *The range analysis of integer variables determines the lowest and highest bounds that each variable assumes throughout the execution of a program. This technique is vital to detect a plethora of software vulnerabilities but the literature does not describe any principled way to apply range analysis on distributed systems. This negligence is unfortunate, as networks are the most common targets of software attacks. The goal of this paper is to set right this omission. Capitalizing on a recent algorithm to infer communication protocols, we have designed, implemented and tested a range analysis for distributed systems. Our contribution, a holistic view of the system, is more precise than analyzing each system module independently. In this paper we support this statement through a number of examples, and experiments performed on top of the SPEC CPU 2006 benchmarks. A prototype of our tool, implemented on the LLVM compiler, is available for scrutiny.*

Resumo. *A análise de largura de variáveis determina o maior e menor valores que cada variável inteira de um programa pode assumir durante a sua execução. Tal técnica é de suma importância para detectar vulnerabilidades em programas mas, até o momento, não existe abordagem que aplique essa análise em sistemas distribuídos. Tal omissão é séria, uma vez que esse tipo de sistema é alvo comum de ataques de software. O objetivo deste artigo é preencher tal lacuna. Valendo-nos de um algoritmo recente para inferir protocolos de comunicação, nós projetamos, implementamos e testamos uma análise de largura de variáveis para sistemas distribuídos. Nosso algoritmo, ao prover uma visão holística do sistema distribuído, é mais preciso que analisar cada parte daquele sistema separadamente. Demonstramos tal fato via uma série de exemplos e experimentos realizados sobre os programas presentes em SPEC CPU 2006. Um protótipo de nossa ferramenta, implementado sobre o compilador LLVM, está disponível para escrutínio.*

1. Introdução

A análise de largura de intervalos [Cousot and Cousot 1977], é uma das técnicas mais importantes que compiladores usam para encontrar vulnerabilidades em programas. Essa análise determina, para cada variável inteira usada em um programa, quais são o menor e o maior valores que ela pode assumir. Tal informação permite ao compilador detectar a possibilidade de ocorrência de dois fenômenos que comprometem a segurança de programas. O primeiro deles é o acesso fora de limites de arranjos. Esse evento ocorre quando

uma variável inteira i indexa um endereço inválido a partir de um ponteiro base a. Erros assim são comuns em linguagens fracamente tipadas, como C, uma vez que a expressão $a[i]$ não assegura que i seja menor que o maior endereço dereferenciável a partir de a. O segundo fenômeno que a análise de largura de variáveis descobre estaticamente são os estouros de inteiros. Uma operação como $j = i + 1$, em linguagens como C, C++ ou Java, pode retornar um valor j menor que i, se i for o maior inteiro representável. Por razões que discutiremos na seção 4, essa semântica pode levar a vulnerabilidades de software.

A análise de largura de variáveis existe há quase 40 anos. Desde a sua concepção original, em 1977 [Cousot and Cousot 1977], vários desafios relacionados à implementação dessa análise foram superados, tanto em termos de precisão [Gawlitza et al. 2009, Su and Wagner 2005], quanto em termos de eficiência [Logozzo and Fahndrich 2008]. Recentemente, por exemplo, cientistas demonstraram como propagar informações de largura de variáveis em estruturas de dados [Oh et al. 2011], eliminando um dos últimos entraves ao projeto de análises de grande precisão. Entretanto, pesquisadores ainda não haviam abordado a análise de largura de variáveis em programas distribuídos. O presente artigo trata desta abordagem.

O grande empecilho à análise de sistemas distribuídos devia-se a um fato simples: até pouco tempo atrás não havia método confiável para determinar, estaticamente, quais operações de envio e recepção de mensagens se comunicam. Em outras palavras, uma vez que operações como `receive`, que coleta mensagens da rede, eram consideradas inseguras, pouco se podia assumir quanto aos valores coletados, já visto que eles podem provir de quaisquer fontes. Entretanto, esse problema foi superado por Teixeira *et al.* [Teixeira et al. 2014] neste ano de 2014. Teixeira *et al.* desenvolveram um algoritmo que infere canais de comunicação entre programas que integram um sistema distribuído. Valendo-nos de tal método, nós projetamos, implementamos e testamos um algoritmo que propaga informações de largura de variáveis entre nós que se comunicam em uma rede.

Nossa solução para o problema da análise de largura de variáveis em sistemas distribuídos consiste em cinco passos. (i) Nós determinamos quais dados representam as mensagens que um programa manipula. (ii) Nós aplicamos a análise de largura de variáveis nesses dados, para determinar o *layout* das mensagens do programa. (iii) Usando as técnicas de Teixeira *et al.*, nós determinamos quais os canais de comunicação existem entre os programas distribuídos. (iv) Nós emparelhamos as mensagens trocadas por esse canal, para determinar quais dados estão fluindo de um programa para o outro. (v) Nós executamos a análise de largura de variáveis uma segunda vez, agora sobre todo o sistema distribuído, obtendo resultados finais. *Dentre esses cinco passos, somente (iii) não é uma contribuição original deste artigo.* Enfatizamos que os passos (i) e (ii) descobrem não somente o *layout* de mensagens, mas o *layout* de arranjos em geral. O fato de podermos entender, de forma automática, como campos estão dispostos em mensagens é uma consequência dessa generalidade. Nossa ferramenta foi implementada sobre o compilador LLVM [Lattner and Adve 2004], e está disponível publicamente.

2. Contextualização

A fim de ilustrar a importância do problema abordado neste artigo, usaremos os dois programas vistos na figura 1. O código mostrado na parte (a) da figura lê uma quantidade `N` de caracteres da entrada padrão, e os envia através de uma conexão de rede, para o programa

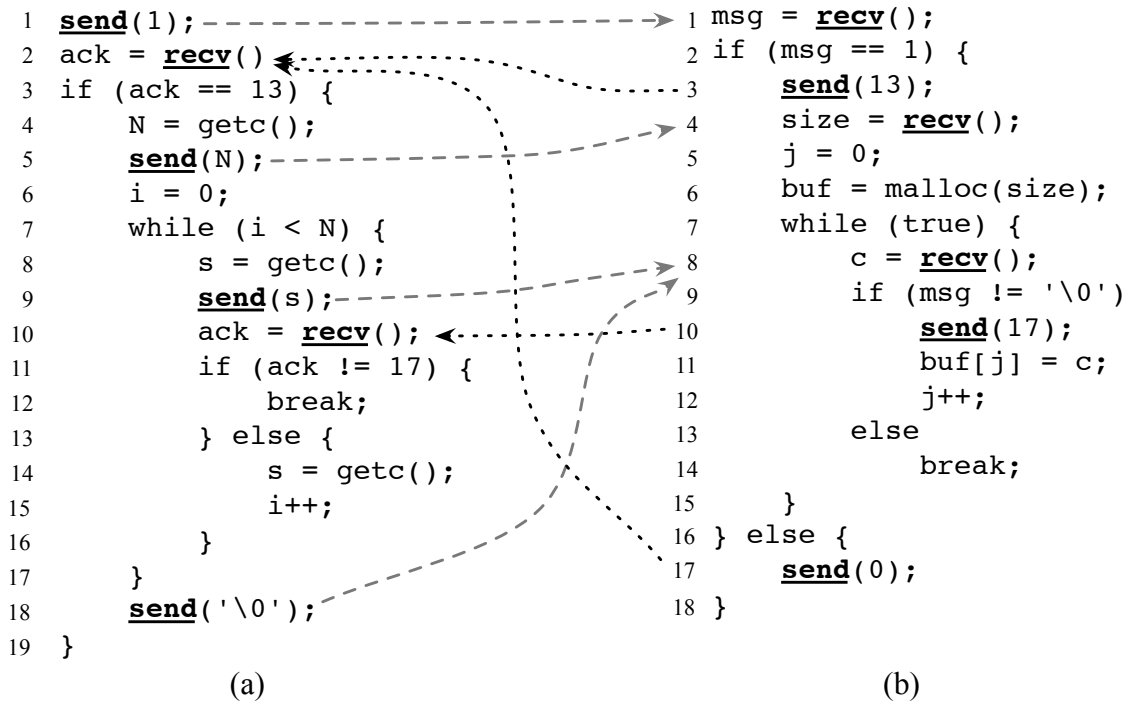


Figura 1. Canais implícitos de comunicação em programa distribuído.

na figura 1 (b). O programa (b) recebe esses caracteres e os armazena em um arranjo de tamanho `N`. Caso fosse possível escrever mais que `N` caracteres no arranjo usado no programa (b), então observaríamos a ocorrência de um fenômeno chamado *buffer overflow*. Esse é um dos principais meios que atacantes usam para comprometer o funcionamento de programas. O nosso programa exemplo, contudo, é seguro: o programa (a) nunca transmite para o programa (b) mais que `N` *bytes* de dados. Entretanto, a análise individual do programa (b) não nos permite inferir tal fato: na ausência de maiores informações, um analisador estático deve, conservadoramente, assumir que o laço na linha 7 daquele programa pode executar mais que `N` iterações.

Cada operação de `send` que se comunica com uma instrução `recv` cria um canal de comunicação *implícito*. A definição de todos os canais implícitos em um sistema distribuído é um problema indecidível. Existem, contudo, algoritmos que apontam a possibilidade de existência de tais canais de forma relativamente precisa. Um deles foi recentemente proposto por Teixeira *et al.*. Os possíveis canais de comunicação implícitos que esse algoritmo encontra aparecem indicados pelas setas tracejadas na figura 1. Analisando esses canais, pode-se concluir que as variáveis `N` e `size` possuem o mesmo valor.

3. Arcabouço de Análises Estáticas

Nossa análise distribuída de largura de variáveis segue a sequência de passos mostrada na figura 2. A análise de segmentação infere os diferentes campos que constituem cada mensagem trocada em um sistema distribuído. A análise local de intervalos nos permite encontrar qual informação está armazenada em cada campo de uma mensagem. A propagação de informações nos diz quais dados fluem de um programa para outro, efetivamente habilitando a última fase de nossa abordagem: a análise global de intervalos.

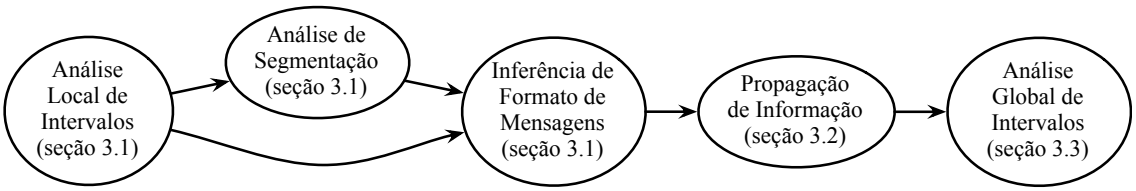


Figura 2. Análises estáticas que compõem o nosso arcabouço de inferência de largura de variáveis em sistemas distribuídos. Todas essas etapas são contribuições deste artigo.

Cada uma dessas análises é discutida nas próximas seções deste artigo.

3.1. Inferência Automática de Formato de Mensagens

O objetivo deste trabalho é portar uma análise de largura de variáveis tradicional para um sistema distribuído. Para alcançar esse objetivo, o primeiro desafio que precisamos vencer é como entender o formato das mensagens trocadas via rede. Normalmente, uma mensagem é implementada como um arranjo. Algumas células desses arranjos são agrupadas em *campos*. Diferentes programas encadeiam esses campos de diferentes maneiras. Campos de mensagens podem conter, por exemplo, seu *opcode*[1], o identificador do remetente, um contador de tempo (costumeiramente conhecido como *time-stamp*), e dados. A figura 3 (a) ilustra um programa que cria um dentre dois tipos diferentes de mensagens, e envia a mensagem criada através de uma operação **send**.

Análise Local de Intervalos. Para inferir como informações são passadas entre nós que se comunicam via rede, utilisamos a mesma análise de largura de variáveis que queremos portar para o mundo distribuído. Porém, dessa vez nós executamos tal análise *localmente*, isto é, de forma individual para cada programa que faz parte do sistema distribuído. A análise de largura de variáveis local nos dá, para cada variável inteira, uma função R, definida da seguinte maneira:

$$R(v) = [l, u], \;\; \{l, u\} \subset \mathbb{Z} \cup \{-\infty, +\infty\}, l \leq u$$

Como existem várias implementações de análises de intervalos descritas na literatura [Cousot and Cousot 1977, Gawlitza et al. 2009, Rodrigues et al. 2013, Su and Wagner 2005], nós omitiremos os detalhes do algoritmo que infere a função R automaticamente. Ao leitor interessado, recomendamos o trabalho de Rodrigues *et al.* [Rodrigues et al. 2013], que descreve uma implementação eficiente de tal análise. Assumiremos, portanto, a existência de uma técnica para construir a função R, que nos informa, para cada variável v, uma estimativa do menor e do maior valores que v assume durante a execução de um programa.

Análise de Segmentação. De posse da função R, passamos à segunda fase de nossa técnica. Nessa etapa, nosso objetivo é inferir para cada ponteiro p uma *tabela de segmentos* que o descreva. A tabela de segmentos associada a um ponteiro p é uma lista de intervalos que podem ser usados para indexar partes de p. Continuando com o nosso exemplo, a figura 3 (b) mostra os intervalos que podem ser usados para indexar o arranjo apontado pelo ponteiro a. Os diferentes ponteiros usados para carregar dados em

[1]O *opcode* de uma mensagem é um valor que descreve o tipo daquela mensagem, e permite ao seu receptor escolher a forma de tratamento mais adequado para ela.

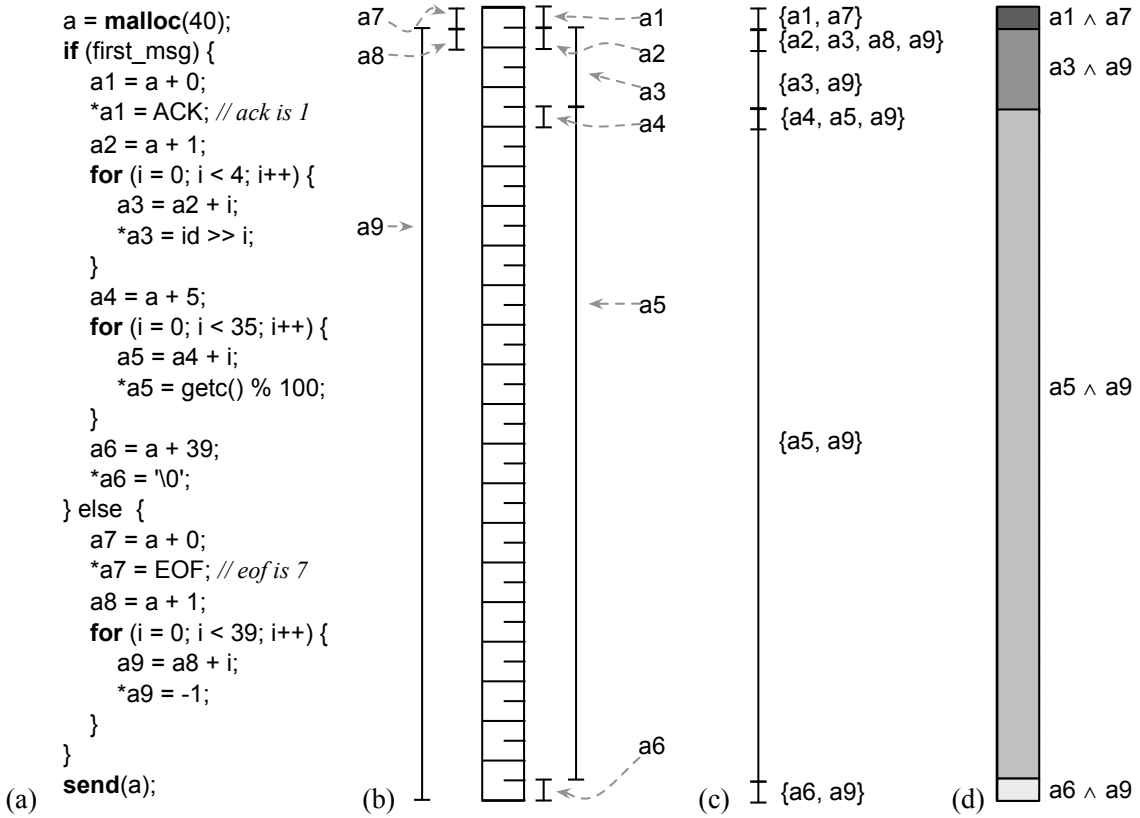


Figura 3. (a) Exemplo de programa que cria e envia mensagens. (b) Os intervalos dos vários ponteiros que podem indexar blocos na mensagem. (c) Outra visão dos intervalos de ponteiros, agrupados por área comum de indexação. (d) A tabela de segmentos da mensagem, formada pelos diferentes ponteiros usados para efetivamente armazenar dados nela via operações de carregamento.

$$
\begin{aligned}
a = \texttt{malloc}(v) \quad &\Rightarrow \quad T(a) = [0, u], \text{ sendo } R(v) = [l, u] \\
a' = a + c \quad &\Rightarrow \quad \begin{cases} R(a') = [\min(l + c, l'), \max(u + c, u')], \\ \qquad\qquad \text{sendo } R(a) = [l, u], R(a') = [l', u'] \\ T(a) = T(a') \end{cases} \\
*a = x \quad &\Rightarrow \quad \text{join}(T(a), R(a))
\end{aligned}
$$

Figura 4. Equações usadas para encontrar os segmentos que constituem cada arranjo de um programa.

um arranjo determinam a sua tabela de segmentos. Assim, a tabela de segmentos para o ponteiro a de nosso exemplo pode ser vista na figura 3 (d). A tabela de segmentos T associada a um arranjo a é construída de acordo com as equações mostradas na figura 4.

A função `merge`, usada para tratar operações de carregamento, e.g., $*a = x$, é definida na figura 5. Nessa implementação, nós usamos a sintaxe de ML, uma linguagem de programação funcional. Tabelas são representadas como listas de tu-


```
fun assertContiguous [] = true
  | assertContiguous [_] = true
  | assertContiguous ((l1, u1, _) :: (l2, u2, p) :: r) =
     u1 = l2 andalso assertContiguous ((l2, u2, p) :: r)

fun merge [] (l, u, p) = [(l, u, [p])]
  | merge ((ll, uu, pp)::r) (l, u, p) =
    if l > uu
    then (ll, uu, pp) :: merge r (l, u, p)
    else if l = ll
       then if u < uu
            then (ll, u, p::pp) :: (u, uu, pp) :: r
            else (ll, uu, p::pp) :: merge r (uu, u, p)
       else if u < uu
            then (ll, l, pp) :: (l, u, p::pp) :: (u, uu, pp) :: r
            else (ll, l, pp) :: (l, uu, p::pp) :: merge r (uu, u, p)


fun join t (l, u, p) = if assertContiguous t
                       then merge t (l, u, p)
                       else nil
```

```
join [(0, 39, [a])]
     (0, 1, a1)
     = [ (0,1,[a1,a]),(1,39,[a]) ]

join [ (0,1,[a1,a]),(1,39,[a]) ]
     (1, 5, a3)
     = [ (0,1,[a1,a]),(1,5,[a3,a]),
         (5,39,[a]) ]

join [ (0,1,[a1,a]),(1,5,[a3,a]),(5,39,[a]) ]
     (1, 5, a9)
     = [ (0,1,[a1,a]),(1,5,[a9,a3,a]),
         (5,39,[a]) ]

join [ (0,1,[a1,a]),(1,5,[a9,a3,a]),
     (5,39,[a]) ]
     (5, 38, a5)
     = [ (0,1,[a1,a]),(1,5,[a9,a3,a]),
         (5,38,[a5,a]),(38,39,[a]) ]
```

Figura 5. Algoritmo que faz o emparelhamento de tabelas de ponteiros. À direita do algoritmo mostramos algumas chamadas da função `join` para os ponteiros vistos na figura 3 (a).

plas. Cada tupla possui três elementos, e.g., (l, u, pp). Os inteiros l e u representam o início e o final do segmento. A lista pp guarda todos os ponteiros que são usados para indexar aquele segmento. O operador ::, em ML, denota a concatenação de listas. Assim, a tabela vista na figura 3 (c) é representada pela seguinte notação: $[(0, 1, [a_1, a_7]), (1, 5, [a_3, a_9]), (5, 38, [a_5, a_9]), (38, 39, [a_6, a_9])]$. Note que os segmentos determinam uma classe de equivalência sobre a tabela. Em outras palavras, cada célula de um arranjo pertence a um segmento e a interseção de dois segmentos diferentes sempre é vazia. Essas propriedades implicam em *contigüidade*, isso é, o intervalo final de um segmento é o intervalo inicial de seu vizinho. Nós salientamos essa propriedade via a função `assertContiguous`, a qual pode ser vista na figura 5. A função `join` recebe uma tabela t e um intervalo para ser inserido em t. Verificada a contigüidade de t, `join` modifica t via uma invocação de `merge`, para que ela passe a conter o novo segmento. As diversas possibilidades de modificação são vistas na parte direita da figura 5.

Quão precisa é nossa análise de segmentação? Neste artigo, estamos interessados em descobrir o *layout* de arranjos usados como mensagens em sistemas distribuídos. Por outro lado, a nossa análise de segmentação é mais geral: ela descobre segmentos em **qualquer** arranjo usado em um programa. Assim, podemos mensurar a precisão da análise contando o número de segmentos descobertos por arranjo: quanto mais segmentos descobrirmos por arranjo, mais precisa é nossa análise de segmentação. A figura 6 mostra esse número para os arranjos nos programas de SPEC CPU 2006. Essa tabela inclui todos os arranjos encontrados naquele *benchmark*. Essa abrangência nos dá uma idéia muito melhor da precisão de nossa análise, que se restringíssemos esse experimento somente aos arranjos trocados como mensagens em sistemas distribuídos, pois esses seriam poucos.

A figura 6 conta o número de tabelas de segmentos, em vez do número de arranjos, pois a mesma tabela pode ser formada por arranjos diferentes. Isso acontece quanto a análise de ponteiros de LLVM não consegue dizer se dois ponteiros podem ou não refe-

Benchmark	#Tabelas	#médio de segmentos	Maior tabela	%Arranjos
433.milc	631	3	34	85.00%
444.namd	617	4	44	72.00%
447.dealII	9,843	2	109	75.00%
450.soplex	2,784	2	55	89.00%
470.lbm	24	16	152	99.00%
401.bzip2	493	2	92	93.00%
429.mcf	103	3	19	68.00%
456.hmmer	4,872	2	64	86.00%
458.sjeng	356	2	14	96.00%
462.libquantum	343	2	6	99.00%
464.h264ref	15,436	2	45	97.00%
471.omnetpp	2,518	2	50	58.00%
473.astar	333	3	22	95.00%
483.xalancbmk	25,194	2	62	82.00%
Total/Média/Max/Média	4,539	3.4	152	85.29%

Figura 6. Precisão de nossa análise de segmentação.

renciar o mesmo endereço base. Pela figura 6, vemos que, em média, cada tabela possui 3.4 segmentos. A maior tabela que observamos, presente em `lbm`, possui 152 segmentos. Em outras palavras, essa tabela representa um arranjo indexado por 152 variáveis contendo intervalos de valores diferentes. A figura 6 contém uma coluna *%Arranjos*, que descreve a porcentagem de arranjos que pudemos analisar por *benchmark*. Somente analisamos arranjos cujo ponteiro base é conhecido. Ou seja, precisamos encontrar, no código do programa, o ponto de criação do arranjo. Arranjos alocados por funções externas[2], por exemplo, não podem ser analisados.

Inferência de Intervalos em Campos de Mensagens. Feita a segmentação da memória nos programas que formam o sistema distribuído, passamos à fase de inferência de valores em mensagens. Nessa etapa, estamos interessados em encontrar quais os intervalos de valores que podem ser armazenados em cada segmento. Fazemos isso via o laço abaixo:

- Para cada instrução de carregamento $*a = x$ presente no programa:
  - Para cada segmento s que contém a in $T(a)$ faça
    - $R(s) = R(s) \cup R(x)$

A figura 7 mostra o resultado da inferência de mensagens quando aplicada no programa visto na figura 3 (a). Inicialmente, todos os segmentos da tabela associada ao arranjo a contém intervalos inteiros indefinidos, indicados pela notação $[?, ?]$. Durante o processamento das instruções de carregamento, os valores desconhecidos são substituídos pelos valores encontrados via a análise local de intervalos. Se duas ou mais instruções, tais como $*a = x_1$ e $*a = x_2$, carregam valores nos mesmos segmentos, então esse segmento recebe a união dos intervalos $R(x_1) \cup R(x_2)$. Esse fenômeno pode ser visto durante o processamento da instrução $*a_7 =$ EOF, na figura 7, que expande o segmento associado à $\{a_5, a_9\}$, de $[0, 99]$ para $[-1, 99]$.

[2]Uma função é *externa* se o seu código fonte não está disponível para nosso compilador.

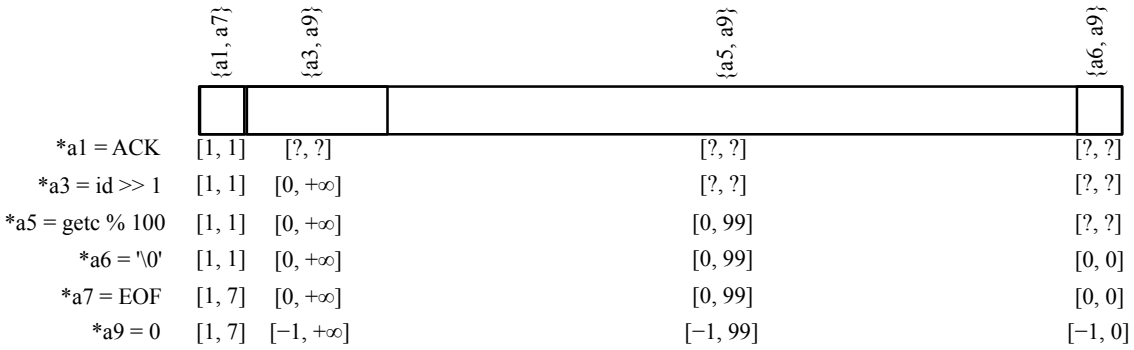


Figura 7. Inferência de intervalos inteiros na mensagem vista na figura 3.

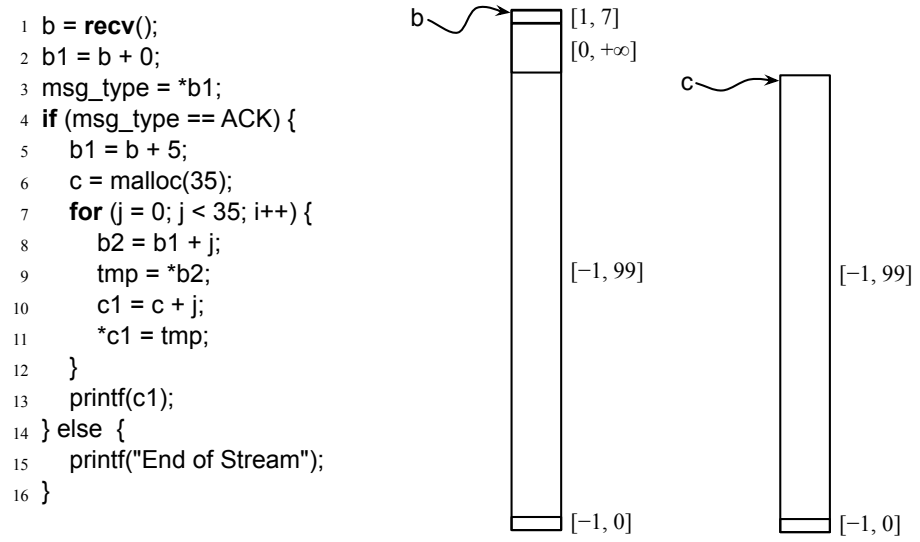


Figura 8. A esquerda, vê-se um trecho de código que recebe mensagens enviadas pelo programa da figura 3 (a). À direita, vêem-se mensagens recuperadas via o emparelhamento por canais implícitos.

A inferência de valores de mensagens termina tão logo todas as instruções de carregamento no programa sejam processadas. Cada instrução é visitada somente uma vez, então a terminação é garantida. Assim, a complexidade computacional dessa análise é proporcional ao número de instruções de carregamento no programa.

3.2. Propagação de Informação entre Programas

Finda a fase local de nosso algoritmo, que envolve os três passos descritos na seção 3.1, passamos à fase distribuída de nossa análise. Nessa etapa, tabelas de segmentos em programas diferentes são emparelhadas, de acordo com os canais de comunicação inferidos usando-se o algoritmo de Teixeira *et al.*. Para cada canal de comunicação inferido, as tabelas enviadas são emparelhadas com as tabelas recebidas. A figura 8 ilustra esse processo. O programa visto nessa figura recebe mensagens enviadas pelas instruções mostradas na figura 3 (a). Uma vez que existe um canal de comunicação implícito entre a instrução **send** da figura 3 (a) e a instrução **recv** da figura 8, temos que as tabelas associadas aos arranjos a e b devem ser emparelhadas.

A partir de um canal de comunicação implícito formado por uma operação de **send** s e uma operação de **recv** r, podemos definir as tabelas de segmentos origem e destino. Chamamos de tabela *origem* aquela associada a qualquer arranjo passado para s e *destino* a tabela associada a algum arranjo recebido por r. Dados esses conceitos, o emparelhamento de tabelas de segmentos é uma operação simples, e consiste no casamento de campos cujos índices se correspondem nas tabelas origem e destino. A figura 8 mostra as tabelas de segmentos associadas aos arranjos b e c, as quais obtemos via emparelhamento com a tabela origem associada ao arranjo a da figura 3 (a).

3.3. Análise Global de Largura de Variáveis

Terminado o emparelhamento, começamos a última fase da abordagem que esse artigo propõe: a análise global de intervalos. Essa etapa não requer qualquer algoritmo especialmente adaptado para o universo dos sistemas distribuídos. Para encontrar os intervalos associados às variáveis inteiras de cada programa que integra o sistema, podemos usar qualquer algoritmo já descrito na literatura. As informações necessárias ao correto funcionamento do algoritmo já foram inferidas nos passos anteriores. Essa flexibilidade é uma das vantagens de nossa abordagem.

Continuando com o nosso exemplo, nós temos que a variável `tmp`, inicializada na linha 9 da figura 8, pode conter somente valores dentro do intervalo $[-1, 99]$. Esse intervalo foi inferido para segmentos apontados pelo ponteiro b_2 no passo de propagação de informação entre nós comunicantes. Caso analisássemos o programa da figura 8 em separado, teríamos de assumir que a variável `tmp` pudesse ser inicializada com qualquer valor inteiro. Essa perda de precisão deve-se ao fato de uma análise individual não nos dar qualquer informação sobre dados recebidos via operações de **recv**.

4. Estudo de Caso

Nós implementamos nossa análise sobre o compilador LLVM, versão 3.3, pois tanto o trabalho de Teixeira *et al.* quanto a análise de largura de variáveis de Rodrigues *et al.* foram construídas nesse compilador. A fim de demonstrar o funcionamento da técnica proposta neste artigo, esta seção descreve sua utilização sobre um par cliente-servidor real. O código presente aqui pode ser compilado e testado diretamente[3]. O cliente usado neste estudo de caso envia para um servidor uma quantidade indeterminada de pares formados por nomes de funcionários e horas trabalhadas. As mensagens que carregam essas informações possuem dois campos. O servidor, ao receber cada um desses pares, multiplica a quantidade de horas trabalhadas por um valor de salário-base e envia de volta para o cliente uma tripla, formada pelo nome do funcionário, suas horas trabalhadas e seu salário. A figura 9 mostra o código de nossa aplicação cliente, e a figura 10 mostra o código de nossa aplicação servidora. Por simplicidade, neste exemplo operaremos somente a nível de *bytes*. Assim, nomes são cadeias de *bytes*, horas trabalhadas são um *byte* e o salário-base é um *byte* também.

A figura 9 mostra, além do programa cliente, o protocolo de comunicação de nosso estudo de caso. O cliente inicialmente informa ao servidor a quantidade de

[3]Devido a restrições de espaço, não mostramos as diretivas `#include` em nosso código. Assim, os seguintes arquivos devem ser incluídos em cada programa: `stdio.h`, `string.h`, `sys/socket.h` e `arpa/inet.h`.


```
1  int main() {
2     int i, num, salario_calc, descriptor, ack = 0;
3     char buffer[42], *message, server_reply[2000];
4     struct sockaddr_in server;
5
6     descriptor = socket(AF_INET , SOCK_STREAM , 0);
7     server.sin_addr.s_addr = inet_addr("localhost");
8     server.sin_family = AF_INET;
9     server.sin_port = htons(2000);
10    connect(descriptor, (struct sockaddr *)&server , sizeof(server));
11
12    // Informa ao servidor quantas mensagens serao enviadas:
13    fscanf(stdin, "%d", &num);
14    buffer[0] = num < 100 ? num : 0;
15    send(descriptor, buffer, 1, 0);
16    recv(descriptor, &ack, sizeof(int), 0);
17    if (ack) {
18       // Envia os pares: (nome do trabalhador x horas trabalhadas):
19       for (i = 0; i < num; i++) {
20          // Leia o nome do trabalhador:
21          fscanf(stdin, "%s", buffer);
22          // Leia o numero de horas trabalhadas:
23          fscanf(stdin, "%d", &buffer[40]);
24          // Permite-se no maximo dez horas-extras trabalhadas:
25          buffer[40] = buffer[40] > 10 ? 10 : buffer[40];
26          // Envia a mensagem:
27          send(descriptor, buffer, strlen(buffer) * sizeof(char), 0);
28          // Recebe o valor a ser pago ao trabalhador:
29          recv(descriptor, buffer, strlen(buffer) * sizeof(char), 0);
30          salario_calc = (int)buffer[41];
31          printf("Valor a ser pago = %d\n", salario_calc);
32       }
33    }
34    return 0;
35 }
```

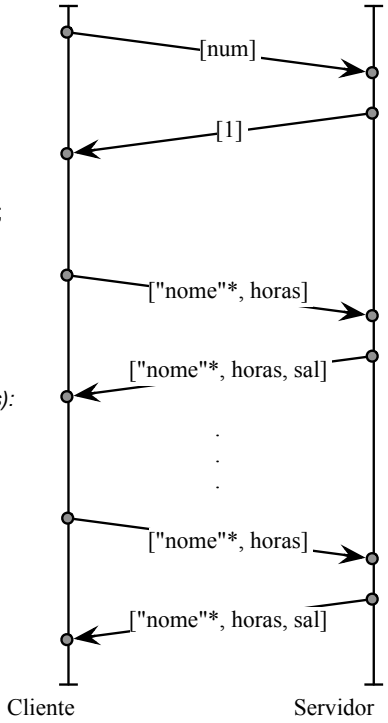


Figura 9. (Esquerda) Programa cliente. (Direita) Protocolo de comunicação.

```
1  int main() {
2     int num, i, salario, descriptor, connection, clilen;
3     char buffer[42];
4     struct sockaddr_in server, client;
5
6     descriptor = socket(AF_INET , SOCK_STREAM , 0);
7     server.sin_addr.s_addr = inet_addr("localhost");
8     server.sin_family = AF_INET;
9     server.sin_port = htons(2000);
10    bind(descriptor, (struct sockaddr *)&server, sizeof(server));
11    connection = accept(descriptor, (struct sockaddr *)&client, &clilen);
12    // Recebe o numero de mensagens que vao chegar:
13    recv(connection, &num, sizeof(int), 0);
14    // Envia um ack para o cliente:
15    buffer[0] = 1;
16    send(connection, buffer, 1, 0);
17    // Leia todas as mensagens:
18    for (i = 0; i < num; i++) {
19       recv(connection, &buffer, strlen(buffer)*sizeof(char), 0);
20       salario = HORA * buffer[40]; // HORA == 12
21       buffer[41] = salario;
22       // Envia o salario final para o cliente:
23       send(connection, buffer, strlen(buffer)*sizeof(char), 0);
24    }
25    return 0;
26 }
```

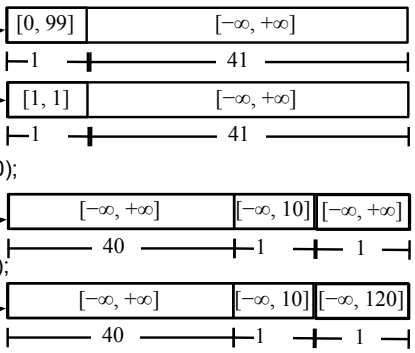


Figura 10. Programa servidor que interage com o cliente visto na figura 9. Os segmentos de mensagens que nossa análise infere automaticamente são mostrados à direita do programa.

mensagens trocadas. Essa primeira mensagem é seguida de uma confirmação de recebimento, por parte do servidor. A partir desse ponto, o cliente passa a enviar os pares (nomes $\times$ horas) para o servidor, que lhe envia de volta as triplas (nomes $\times$ horas $\times$ salário). Esse protocolo possui quatro canais implícitos de comunicação[4]: (i) `cliente:send:15` $\rightarrow$ `servidor:recv:13`, (ii) `servidor:send:16` $\rightarrow$ `cliente:recv:16`, (iii) `cliente:send:27` $\rightarrow$ `servidor:recv:19` e (iv) `servidor:send:23` $\rightarrow$ `cliente:recv:29`. O algoritmo proposto por Teixeira *et al.* descobre esses quatro canais e somente eles. Note que nesse caso, o algoritmo não reporta falsos-positivos: todos os canais inferidos são, de fato, canais válidos.

O algoritmo proposto na seção 3.1 descobre os formatos de mensagens vistos à direita da figura 10. Nesse exemplo, nosso algoritmo foi capaz de inferir, de forma precisa, os intervalos de valores associados a cinco dos dez campos de mensagens trocadas entre cliente e servidor. Por exemplo, o programa cliente possui um teste na linha 25 (figura 9) que garante que trabalhadores não podem fazer mais que 10 horas extra. Esse teste permite-nos determinar que somente valores no intervalo $[-\infty, 10]$ podem ser transferidos na posição 40 das mensagens. Valores imprecisos, associados aos intervalos $[-\infty, +\infty]$, devem-se à pouca informação disponível no código fonte do programa. São imprecisos, por exemplo, os valores dos *bytes* associados a nomes de funcionários. Esses *bytes* podem cobrir qualquer intervalo entre $[-128, 127]$, exatamente o domínio do tipo `char`.

4.1. Como utilizar nossa análise para aumentar a segurança e a eficiência de programas distribuídos.

Estamos, atualmente, usando os resultados de nossa análise para eliminar guardas sobre operações que podem causar estouro em aritmética de inteiros. Linguagens como C, C++ ou Java tratam operações inteiras segundo uma semântica modular. Se o resultado de uma instrução inteira for maior que o tamanho do registrador onde esse resultado será armazenado, então os *bits* mais significativos desse valor são descartados. Por exemplo, $6_{char} \times 22_{char} = -124_{char}$. A literatura contém várias descrições de ataques baseados nesse semântica [Brumley et al. 2007, Dietz et al. 2012].

Um ataque baseado em estouro de arranjos é possível em nosso estudo de caso. Considere, por exemplo, que um usuário malicioso informe um valor negativo de horas na linha 23 do programa cliente (variável `buffer[40]` na figura 9). Suponhamos que tal valor seja o inteiro negativo -75. Temos então que o teste na linha 25 do programa cliente é falso. Consequentemente -75 será transmitido para o servidor. No código do servidor (figura 10), a multiplicação na linha 20, e.g., $-75_{char} \times 12_{char} = 124_{char}$, produz um valor maior que o máximo número de horas cuja intenção do desenvolvedor seria permitir, isto é, $10_{char} \times 12_{char} = 120_{char}$. Esse tipo de falha de segurança é difícil de ser detectado sem o auxílio de ferramentas de análise estática, e pode levar a situações catastróficas. A título de exemplo, em 1996, o foguete Ariane 5 foi perdido devido a um estouro de inteiros – tal erro de software custou ao programa espacial europeu cerca de US$ 370 milhões [Dowson 1997].

Existem diversas técnicas para sanear programas contra estouros de operações inteiras. Recentemente, por exemplo, Rodrigues *et al.* propuseram um gerador de código que instrumenta operações inteiras em um programa. Essa instrumentação invoca código

[4]Números ao lado do nome do programa denotam linhas nas figuras 9 e figuras 10.

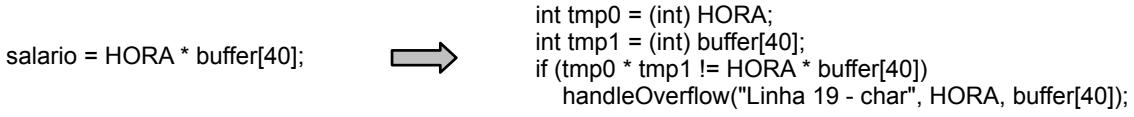


Figura 11. Exemplo de código para verificar se houve estouro de inteiro

de tratamento de erros sempre que um estouro é detectado. Tal técnica, quando aplicada ao programa da figura 10, insere guardas na soma da linha 18, e na multiplicação da linha 20. Cada uma dessas guardas é implementada como uma combinação de testes condicionais e instruções de desvio, conforme mostrado na figura 11.

Essas guardas tornam o programa instrumentado mais lento que o programa original. Conforme reportado por Dietz *et al.*, essa lentidão pode comprometer até 15% do tempo de execução do programa modificado [Dietz et al. 2012]. Nossa técnica nos permite eliminar alguns desses testes, e também indicar ao desenvolver quais testes precisam ser mantidos. Por exemplo, considerando-se o programa servidor, visto na figura 10, nossa análise elimina o teste sobre o incremento realizado na linha 18, pois a variável `i` é limitada por `num`, cujo intervalo superior pode ser no máximo 99. Por outro lado, não podemos eliminar a guarda da multiplicação da linha 20, pois `buffer[40]`, caso fosse um número negativo muito pequeno, causaria um estouro aritmético. Nossa análise detecta também essa possibilidade e mantém a instrumentação na linha 20.

5. Trabalhos Relacionados

O presente trabalho relaciona-se a pesquisa desenvolvida tanto em análise de código, quanto em sistemas distribuídos. No primeiro caso, nossa inspiração mais importante deve-se a Cousot e Cousot [Cousot and Cousot 1977], que introduziram o conceito de análise de largura de variáveis. No segundo caso, contudo, nossa inspiração é bem mais recente: muito do que discutimos neste artigo foi possível somente devido ao arcabouço construído por Teixeira *et al.*. No restante dessa seção discutiremos como nosso trabalho se relaciona com outras pesquisas nesses dois campos.

Análise de largura de variáveis. A análise de largura de variáveis é um dos exemplos clássicos de interpretação abstrata. A técnica de interpretação abstrata, introduzida por Cousot e Cousot, é um arcabouço teórico que permite a compiladores obter informações de um programa, garantindo que os algoritmos usados terminam. Existem muitas formas de se implementar análise de largura de intervalos [Gawlitza et al. 2009, Mahlke et al. 2001, Stephenson et al. 2000, Su and Wagner 2005]. Essas técnicas seguem duas avenidas principais, que, embora levem ao mesmo objetivo, atravessam caminhos muito diferentes. As técnicas mais conhecidas, como o trabalho de Stephenson *et al.* [Stephenson et al. 2000] ou Mahlke [Mahlke et al. 2001], baseiam-se em algoritmos iterativos. Em outras palavras, esses métodos interpretam as instruções de um programa abstratamente. O programa é interpretado de forma que o valor abstrato, isso é, o intervalo, associado a cada variável inteira somente cresce. Um operador especial, conhecido como alargamento, assegura que esse crescimento termina após algumas iterações. Existem implementações desses algoritmos em compiladores industriais, como Open64 ou LAO, usado pela companhia STMicroelectronics. A maior parte dos artigos acadêmicos, contudo, descrevem algoritmos que resolvem a análise de largura de

intervalos de forma não iterativa. Entre esses trabalhos, citam-se as técnicas de Su e Wagner [Su and Wagner 2005], Gawlitza *et al.* e Rodrigues *et al.*. Exceto o algoritmo de Rodrigues *et al.*, presente em LLVM, não sabemos de outras implementações de algoritmos não iterativos em compiladores de uso industrial.

Em que nosso trabalho difere das técnicas já existentes. O foco deste artigo não é em algoritmos de largura de variáveis *per se*. Nós queremos aplicar tais técnicas em sistemas distribuídos. Com tal propósito, podemos utilizar qualquer algoritmo existente. Neste trabalho usamos o método de Rodrigues *et al.*. Nossa escolha foi motivada por razões puramente pragmáticas: esse método já estava implementado sobre o compilador LLVM, o qual usamos em nossos experimentos. Nessa flexibilidade, conforme já mencionamos antes, reside muito da beleza de nossa abordagem: as técnicas descritas neste artigo, como a propagação de valores entre nós de programas distribuídos, a inferência de formatos de mensagens e a associação de valores abstratos a campos de mensagens são ortogonais à técnica de largura de variáveis usada.

Análise de sistemas distribuídos. Teixeira *et al.* introduziram o algoritmo que usamos para encontrar canais implícitos em sistemas distribuídos. Aquele trabalho identifica canais de comunicação com base nos comandos de rede. A partir desses comandos, Teixeira *et al.* realizam a interconexão dos grafos de controle de fluxo de cada programa. Existem outros trabalhos desenvolvidos com propósito semelhante. Entre eles, destacamos Kleenet, de Sasnauskas *et al.* [Sasnauskas et al. 2010] e T-Check, de Li *et al.* [Li and Regehr 2010]. Essas ferramentas executam o sistema simbolicamente a procura de defeitos de software e permitem a exploração automática de caminhos de execução em aplicações distribuídas. Se uma asserção falha, essas ferramentas registram o caso de teste para que o cenário possa ser repetido.

Em que nosso trabalho difere das técnicas já existentes. O trabalho de Teixeira *et al.* propõe um arcabouço para a análise de sistemas distribuídos, mas não implementa qualquer análise sobre ele. Os autores daquele projeto não tiveram, por exemplo, de lidar com o *layout* de mensagens. Tampouco foi essa uma preocupação de Sasnauskas *et al.* e Li *et al.*. Nesses dois casos, o desenvolvedor deve marcar variáveis para serem simbólicas e deve escrever asserções sobre o estado do sistema, ou seja, usuário deve, explicitamente, indicar ao analisador estático como dados trafegam em mensagens. Esse passo é manual e requer conhecimento sobre a lógica da aplicação e estruturas de dados. Assim, a complexidade da solução depende das entradas simbólicas e do número de nós. Os autores de Kleenet, por exemplo, reportaram que mesmo com entradas simbólicas pequenas e poucos nós, algumas aplicações geram milhares de caminhos de execução. Nossa abordagem é mais automática: o desenvolvedor indica quais funções fazem a comunicação de rede (passo também necessário para os trabalhos relacionados) e o compilador descobre como os dados são passados, sem qualquer intervenção do usuário.

6. Conclusão

Este artigo descreveu uma forma de inferir a largura de variáveis em programas distribuídos. Essa técnica dá a uma ferramenta de análise de código mais subsídios para encontrar vulnerabilidades em aplicações distribuídas. Demonstramos esse fato mostrando como nossa análise nos permite proteger código contra vulnerabilidades devido a estouro de operações aritméticas em valores inteiros. Nossa técnica exige mínima intervenção do

usuário, a saber, a indicação de quais funções fazem acesso à rede. Como trabalho futuro, pretendemos usar o arcabouço descrito neste artigo para sanear programas contra ataques de estouro de *buffer*. Estamos já trabalhando ativamente para alcançar tal objetivo.

Agradecimentos Este projeto é financiado pela Intel, pelo CNPq e pela FAPEMIG.

Referências

Brumley, D., Song, D. X., cker Chiueh, T., Johnson, R., and Lin, H. (2007). RICH: Automatically protecting against integer-based vulnerabil ities. In *NDSS*. USENIX.

Cousot, P. and Cousot, R. (1977). Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252. ACM.

Dietz, W., Li, P., Regehr, J., and Adve, V. (2012). Understanding integer overflow in c/c++. In *ICSE*, pages 760–770. IEEE.

Dowson, M. (1997). The ariane 5 software failure. *SIGSOFT*, 22(2):84–.

Gawlitza, T., Leroux, J., Reineke, J., Seidl, H., Sutre, G., and Wilhelm, R. (2009). Polynomial precise interval analysis revisited. *Efficient Algorithms*, 1:422 – 437.

Lattner, C. and Adve, V. S. (2004). LLVM: A compilation framework for lifelong program analysis & transformation. In *CGO*, pages 75–88. IEEE.

Li, P. and Regehr, J. (2010). T-check: Bug finding for sensor networks. In *IPSN*, pages 174–185.

Logozzo, F. and Fahndrich, M. (2008). Pentagons: a weakly relational abstract domain for the efficient validation of array accesses. In *SAC*, pages 184–188. ACM.

Mahlke, S., Ravindran, R., Schlansker, M., Schreiber, R., and Sherwood, T. (2001). Bitwidth cognizant architecture synthesis of custom hardware accelerators. *TCADICS*, 20(11):1355–1371.

Oh, H., Brutschy, L., and Yi, K. (2011). Access analysis-based tight localization of abstract memories. In *VMCAI*, pages 356–370. Springer.

Rodrigues, R. E., Campos, V. H. S., and Pereira, F. M. Q. (2013). A fast and low overhead technique to secure programs against integer overflows. In *CGO*. ACM.

Sasnauskas, R., Landsiedel, O., Alizai, M. H., Weise, C., Kowalewski, S., and Wehrle, K. (2010). Kleenet: discovering insidious interaction bugs in wireless sensor networks before deployment. In *IPSN*, pages 186–196. ACM.

Stephenson, M., Babb, J., and Amarasinghe, S. (2000). Bitwidth analysis with application to silicon compilation. In *PLDI*, pages 108–120. ACM.

Su, Z. and Wagner, D. (2005). A class of polynomially solvable range constraints for interval analysis without widenings. *Theoretical Computer Science*, 345(1):122–138.

Teixeira, F., Pereira, F., Viera, G., Marcondes, P., Wong, H. C., and Nogueira, J. M. (2014). Siot: defendendo a internet das coisas contra exploits. In *SBRC*, pages 85–96. SBC.

Controlando a Frequência de Desvios Indiretos para Bloquear Ataques ROP

Mateus Tymburibá Ferreira, Ailton Santos Filho, Eduardo Feitosa

[1]IComp/UFAM, Manaus, Brasil

mateustymbu@gmail.com, ailton.santos07@gmail.com, efeitosa@icomp.ufam.edu.br

Abstract. *Because of its wide use in attacks against modern computing systems, protections against malicious codes based on the technique called Return-Oriented Programming (ROP) have been extensively studied. Nevertheless, it is not yet known a definitive solution. This article demonstrates that by controlling the frequency of indirect branch instructions it is possible to avoid the consolidation of ROP attacks. For this, we developed a prototype for Linux, Windows, OSX and Android environments. Experiments conducted with exploits confirmed the effectiveness of the proposed model at a comparable and, in some cases, lower computational cost than that achieved by related protections.*

Resumo. *Em função de seu vasto emprego em investidas contra sistemas computacionais modernos, proteções contra códigos maliciosos baseados na técnica denominada Return-Oriented Programming (ROP) têm sido extensamente estudadas. Apesar disso, ainda não se conhece uma solução definitiva. Este artigo demonstra que, através do controle da frequência de instruções de desvio indireto, é possível evitar a consolidação de ataques ROP. Para isso, foi desenvolvido um protótipo destinado a ambientes Linux, Windows, Android e OSX. Experimentos realizados com exploits confirmaram a eficácia do modelo proposto a um custo computacional comparável e, em alguns casos, inferior àquele alcançado por proteções correlatas.*

1. Introdução e Motivação

Nas últimas décadas, o software tornou-se o elo mais fraco da cadeia de componentes alvejados por atacantes em atos hostis contra sistemas computacionais [Hoglund and McGraw 2004]. Dentre as variadas formas de exploração para execução de código arbitrário, a técnica denominada *Return-Oriented Programming* (ROP) tem despertado grande interesse da comunidade científica e da indústria de segurança de sistemas, em função da sua larga utilização em ataques recentes a sistemas computacionais. Empregos da técnica ROP em ataques bem sucedidos podem ser vistos, por exemplo, nos *malwares* Stuxnet e Duqu [J. Callas 2011] e no código usado na violação de um tipo de urna de votação eletrônica empregada em diversas localidades [Checkoway et al. 2009].

Por ter se tornado uma das principais técnicas utilizadas por atacantes para desenvolver *exploits*[1], mitigações contra o ROP têm sido amplamente estudadas. Contudo, ainda não há uma solução definitiva. O Windows 8, por exemplo, agrega

[1] *Exploit*: artefato desenvolvido com a finalidade de explorar uma vulnerabilidade presente em um sistema

um novo mecanismo de proteção contra o ROP, que impede a chamada de APIs (*Application Programming Interfaces)* tipicamente utilizadas em ataques ROP, caso os parâmetros não estejam armazenados na área de pilha do processo. No entanto, poucos dias depois do lançamento da versão preliminar do sistema, pesquisadores apresentaram demonstrações de estratégias relativamente simples capazes de burlar essa defesa [D. Rosenberg 2011, N. H. Son 2011].

Diante desse contexto, este trabalho tem por objetivo apresentar uma nova estratégia para detecção e bloqueio de ataques ROP: o controle da frequência de instruções de desvio indireto. A eficácia dessa solução no bloqueio de *exploits* é demonstrada através da construção de um protótipo, testado com códigos maliciosos disponíveis no repositório público Exploit Database[2]. A análise de desempenho da solução também é avaliada e comparada com soluções correlatas.

As contribuições deste trabalho são duas: a elaboração e demonstração da eficácia do controle da frequência de desvios indiretos como estratégia para detecção de ataques ROP, incluindo suas variantes inexploradas pela maioria das soluções atuais; e o desenvolvimento de um protótipo de proteção contra ataques ROP destinado a ambientes Windows, Linux, OSX e Android em um *framework* de instrumentação binária dinâmica.

2. Return-Oriented Programming

Tão logo proteções de memória começaram a ser incorporadas aos sistemas computacionais (pilha não executável[3] e bit de execução[4], por exemplo), surgiram novas propostas de ataques alternativos baseados no reaproveitamento dos códigos originais das aplicações [S. Designer 1997].

Nos primeiros ataques de reúso de código, a biblioteca padrão de C (libc) foi o alvo dos desvios do fluxo de execução, os chamados ataques *return-into-libc* (RILC). Contudo, como qualquer código disponível, tanto no segmento de código executável do programa quanto na área de instruções pertencente a uma outra biblioteca carregada, pode ser utilizado, surgiram ideias para interligar funções [J. McDonald 1999, Wojtczuk 2001] ou trechos de código executáveis [T. Newsham 1997, S. Krahmer 2005]. Ao ser demonstrado que o encadeamento desses trechos de código permite a execução de computações arbitrárias (*Turing complete computation*) [Shacham 2007], essa técnica se popularizou entre atacantes e passou a ser referida por *Return-Oriented Programming* ou ROP.

Ao contrário da tradicional RILC, na qual o atacante desvia o fluxo de execução para o início de alguma função útil para o ataque, o ROP encadeia vários pequenos trechos de código (*gadgets*) a fim de executar uma determinada tarefa. Para conseguir esse encadeamento, a última instrução de cada trecho de

[2]http://www.exploit-db.com/

[3]Pilha não executável (*non-executable stack* ou *nx-stack*) é um mecanismo que impede a inserção e execução de instruções (código malicioso) oriundas da área de pilha. Atualmente, esse tipo de proteção está presente por padrão na maioria dos sistemas operacionais.

[4]Bit de execução (NX/XD) é uma extensão natural do mecanismo de pilha não executável concebida para bloquear esse tipo de tentativa em outras áreas da memória. Essa estratégia baseia-se na utilização de um recurso incorporado aos processadores, em 2004, para marcar as páginas de memória com um bit de execução.

código escolhido deve executar um desvio. A ideia original do ROP utiliza *gadgets* finalizados com instruções de retorno (RET) para interligar as frações de código escolhidas [Shacham 2007]. Posteriormente, novos trabalhos demonstraram a possibilidade de utilização de instruções do tipo jump indireto (JMP) para encadear os *gadgets*, ataque que foi batizado de JOP (*Jump-Oriented Programming*) [Chen et al. 2011, Checkoway et al. 2010]. Apesar de não ter sido demonstrado que é possível executar computações arbitrárias (*Turing complete computations*) interligando apenas *gadgets* terminados com a instrução do tipo chamada de função (CALL), essas instruções de desvio indireto também apresentam a capacidade de interligar *gadgets* e podem, portanto, ser utilizadas em ataques ROP [Checkoway et al. 2010].

A técnica de desenvolvimento de *exploits* ROP baseia-se no reúso de código para superar a proteção oferecida pelo bit de execução (NX/XD). Porém, a construção de *shellcodes*[5] inteiramente formados por *gadgets*, usando a técnica ROP, pode ser muito custosa e até inviável, dependendo da disponibilidade de *gadgets* na área de memória executável do processo atacado. Por isso, usualmente as cadeias de *gadgets* existentes nos *exploits* ROP limitam-se à função de preparar o ambiente para a posterior execução do *shellcode*. Esse encadeamento de códigos efetuado nos *malwares* ROP antes de desviar o fluxo de execução para o *shellcode* pode ter como objetivo realizar diversas tarefas: habilitar o bit de execução para a região de memória onde o *shellcode* se localiza, copiar o *shellcode* para uma área de memória com permissão de execução ou desabilitar a proteção oferecida pelo bit NX/XD.

Maiores explicações sobre o funcionamento de ataques ROP, incluindo exemplos, podem ser encontrados em [Ferreira et al. 2012].

3. Trabalhos Relacionados

Normalmente, as sequências de instruções que compõem cada *gadget* usado em um ataque ROP são extremamente curtas, dificilmente contendo mais do que cinco instruções. Essa é uma característica inerente aos ataques ROP, porque quanto maior a sequência de instruções, maior a probabilidade de existir entre essas instruções uma operação que altere o status da memória ou de um registrador de forma a comprometer o ataque.

Diante dessa constatação, diversos autores investiram esforços em uma estratégia de controle da frequência de instruções de retorno como forma de detectar a execução de cadeias de *gadgets*. Foram propostos trabalhos que verificam o pico na frequência de instruções de retorno através do monitoramento em tempo real de cada instrução de retorno executada [Davi et al. 2009, Bania 2010]. Outras soluções adotaram a postura de contabilizar a frequência de instruções de retorno previamente executadas, através da análise de um *buffer* de desvios disponível em hardware (*Branch Trace Store buffer*) [Yuan et al. 2011], ou da verificação da distância entre os endereços de retorno armazenados na área mais recentemente desocupada da pilha [Min et al. 2013, Jiang et al. 2011].

Apesar de eficaz no bloqueio aos ataques ROP identificados até o momento, da forma como vem sendo empregada, a estratégia de controle da frequência de

[5]Shellcode é um conjunto de instruções que, ao serem executadas pelo processador, efetuam alguma atividade maliciosa.

instruções de retorno apresenta as seguintes deficiências:

- Uma sequência de retornos de funções próximos, situação típica em funções com recursão em cauda[6], pode induzir proteções baseadas na estratégia de controle da frequência de instruções de retorno a bloquear equivocadamente a execução de códigos autênticos.
- No caso das soluções que analisam a frequência de instruções de retorno percorrendo a pilha, o atacante pode forjar pilhas estruturadas com valores quaisquer entre os endereços de retorno a fim de superar essa proteção. Basta que os *gadgets* possuam alguma instrução que incremente o registrador ESP (ponteiro de topo de pilha), por exemplo.
- Nesse mesmo cenário (análise da proximidade de endereços de retorno na pilha), funções que contenham poucas variáveis e parâmetros podem apresentar endereços de retorno próximos, levando à ocorrência de falsos positivos. Essa possibilidade pode ainda aumentar caso tenha sido utilizada a otimização de compilação que emprega o registrador EBP como um registrador de uso geral, pois isso força a liberação dos espaços na pilha reservados para armazenar *frame pointers*.

4. Controle da Frequência de Desvios Indiretos

Diante da constatação de que os ataques ROP obrigatoriamente apresentam uma elevada concentração de instruções de desvio indireto (RETs, JMPs indiretos ou CALLs indiretos) em um curto espaço de tempo, a solução proposta neste trabalho é focada no controle da frequência de instruções de desvio indireto com o intuito de detectar as três variantes desse tipo de ataque. Assim, ao invés de medir a frequência apenas das instruções de retorno, esse novo esquema supervisiona a frequência de qualquer tipo de desvio indireto, incluindo aqueles efetuados através de instruções CALL ou JMP indireto, o que possibilita evitar os três tipos de ataques ROP.

O esquema proposto consiste em checar se a contagem do número de instruções de desvio indireto em uma determinada 'janela de instruções' é maior do que um determinado limiar. Para definir o valor ideal desse limiar, é possível tanto estabelecer um valor universal, com base na análise de um conjunto de aplicações, quanto efetuar uma etapa de treinamento com cada software que se pretende proteger, a fim de estabelecer o limiar máximo atingido por aquela aplicação durante a sua execução. É importante ressaltar que a adoção de um limiar específico é melhor do que o uso de um limiar padrão, porque ela impõe restrições mais severas para a construção de uma cadeia de *gadgets* capaz de iludir a proteção.

Inicialmente, pode-se imaginar que existirão muitos casos de aplicações cujas execuções normais apresentem uma elevada densidade de desvios indiretos, em função da execução de laços (*loops*) para repetição de instruções. Contudo, é importante lembrar que - fora as instruções de retorno - as instruções de desvio indireto são raramente utilizadas, restringindo-se a situações muito específicas, como chamadas de funções virtuais (em linguagens orientadas a objetos), estruturas de controle do tipo 'switch-case', chamadas para ponteiros de funções e chamadas de funções pertencentes a bibliotecas ligadas dinamicamente. A seguir, são analisados os três

[6]Funções recursivas em cauda são aquelas nas quais a chamada recursiva é a última instrução a ser executada.

cenários de aplicações autênticas cujas frequências de desvios indiretos teoricamente mais se aproximam da frequência tipicamente registrada durante ataques ROP.

4.1. Laços de repetição com estrutura de controle

Nos tradicionais laços de execução, são as instruções de desvios condicionais que garantem a repetição do corpo do laço, dependendo das condições de parada. Na arquitetura x86, todas as instruções de salto condicional possuem um endereço imediato, registrado na própria instrução. Desvios indiretos são inseridos dentro de laços de repetição apenas quando existem estruturas de controle, como 'if/else', ou chamadas a procedimentos. O impacto desses dois casos na frequência de desvios indiretos foi analisado isoladamente.

Quando uma estrutura de controle, como 'if/else', exige a execução de um salto para uma posição cuja distância em relação ao contador de programa (*Program Counter* - PC) é maior do que é possível representar na instrução, é utilizada uma instrução de salto indireto. Como na arquitetura x86 os valores imediatos podem ser de 8, 16 ou 32 bits, é possível efetuar um desvio direto para uma distância entre o destino do salto e a instrução de desvio de até 2 GB. No entanto, apesar de pouco usual por fugir do padrão, podem existir códigos customizados em que o corpo de um laço de repetição inclua uma instrução de desvio indireto.

Para avaliar o impacto desse tipo de situação na frequência de desvios indiretos, foi analisado um exemplo de código que efetua um 'laço mínimo'. Esse laço é considerado mínimo porque possui apenas a estrutura necessária para analisar a condição de repetição do laço, além da estrutura de controle 'if/else', responsável pela inserção de um desvio indireto (JMP). A Figura 1 apresenta, à esquerda, um exemplo de código de um laço mínimo escrito na linguagem C e, à direita, o código assembly equivalente. O código assembly está representado segundo a sintaxe adotada pela Intel [Universitet 2014].

```
int main(){
   int i=0;
   // executa loop que não faz nada
   do {
      // "if/else" força a inserção de um JMP
      if(i < 1000){i++;}
      else{i++;}
   } while (i < 1000);
   return(0);
}
```

```
main:
  push ebp
  mov  ebp, esp
  sub  esp, 4
  mov  DWORD PTR [ebp-4], 0
.L4:
  cmp  DWORD PTR [ebp-4], 999
  jg   .L2
  add  DWORD PTR [ebp-4], 1
  jmp  .L3
.L2:
  add  DWORD PTR [ebp-4], 1
.L3:
  cmp  DWORD PTR [ebp-4], 999
  jle  .L4
  mov  eax, 0
  leave
  ret
```

Figura 1. Código de laço mínimo

Ao analisar o código assembly, constata-se que a instrução JMP será executada a cada 6 instruções, frequência três vezes menor do que a média observada nas cadeias de *gadgets* presentes nos *exploits* ROP catalogados neste trabalho. Na prática, para tornar o laço de repetição útil, seria incluída, pelo menos, mais uma instrução de máquina, já que o laço mínimo apresentado apenas incrementa a variável que controla a condição de repetição. Isso reduziria ainda mais a frequência de desvios indiretos, o que permite concluir que a eventual execução de instruções de desvio indireto dentro de laços de repetição não acarreta em situações de falso positivo com a solução proposta.

4.2. Funções recursivas

Outra situação que pode gerar uma alta densidade de desvios indiretos é a frequente chamada a procedimentos. Funções com recursividade no início podem gerar uma alta densidade de desvios do tipo CALL, enquanto aquelas com recursividade em cauda podem acarretar em uma elevada frequência de instruções de retorno. Novamente, apesar de não ser comum a existência de funções recursivas que efetuam a chamada recursiva através de uma instrução de CALL indireto, códigos desenvolvidos manualmente em linguagem de montagem podem, eventualmente, fugir do padrão. Em função disso, assim como na análise de um laço de repetição mínimo, foi desenvolvido um exemplo de função recursiva mínima, que executa apenas a checagem da condição de fim da recursão. Os códigos que representam essa função recursiva mínima nas linguagens C e assembly estão indicados na Figura 2.

```
void recursao_minima(int i){
   // condição de parada da recursão
   if(i>0){
      // chamada recursiva
      recursao_minima(i-1);
   }
   return;
}

int main(){
  //inicia chamada recursiva de função
   recursao_minima(1000);
   return(0);
}
```

```
recursao_minima:
   sub  esp, 4
   cmp  DWORD PTR [esp+8], 0
   jle  .L1
   mov  eax, DWORD PTR [esp+8]
   sub  eax, 1
   mov  DWORD PTR [esp], eax
   call recursao_minima
.L1:
   add  esp, 4
   ret

main:
   sub  esp, 4
   mov  DWORD PTR [esp], 1000
   call recursao_minima
   mov  eax, 0
   add  esp, 4
   ret
```

Figura 2. Código de função recursiva mínima

Para simular o cenário com a maior frequência de desvios possível, foi utilizada a otimização de compilação que omite o ponteiro de *frame*, liberando o registrador EBP para uso geral. O uso desse tipo de otimização implica no descarte das instruções *PUSH EBP* e *MOV EBP, ESP*, que tradicionalmente aparecem no início do código de uma função. Em chamadas sucessivas a essas funções, a remoção dessas instruções pode impactar significativamente na frequência de instruções de desvio indireto. A omissão do ponteiro de *frame* acarreta também na substituição da instrução *LEAVE* pela instrução *ADD ESP, valor*, mas essa alteração não repercute em mudanças na densidade de instruções de salto indireto. Ao analisar o código assembly da Figura 2, constata-se que a instrução CALL será executada a cada 7 instruções. Se a otimização de compilação que omite o ponteiro de frame não for utilizada, essa relação passa para uma instrução de chamada de procedimento a cada 9 instruções. Na prática, existirão outras instruções dentro da função para torná-la útil. Portanto, constata-se que a execução sucessiva de instruções CALL no início de funções recursivas não acarreta em falsos positivos.

O mesmo não ocorre para funções com recursividade em cauda, uma vez que a frequência de desvios pode atingir um salto a cada 2 instruções executadas, caso a omissão do ponteiro de *frame* seja empregada, e entre 1 e 3 com o uso do ponteiro. Para evitar esse tipo de erro, pode-se utilizar um mecanismo para identificar a ocorrência de recursões em cauda. No entanto, uma vez que nenhuma ocorrência de falso positivo foi identificada durante os experimentos, esse mecanismo não está descrito neste artigo.

4.3. Laços de repetição com chamada de função

A fim de verificar a possibilidade de ocorrência de falsos positivos, analisou-se ainda um terceiro exemplo de código. Trata-se de um laço de repetição com uma chamada interna para um procedimento com poucas instruções. A Figura 3 apresenta exemplos de código para essa situação na linguagem C e em assembly, considerando-se a omissão do ponteiro de *frame*. No exemplo ilustrado, a função chamada durante a execução do laço de repetição executa uma única operação, responsável por incrementar uma variável global.

```
int cont=0; // variável global

// apenas incrementa a variável global
void contador(){
   cont++;
   return;
}

int main(){
  int i=0;
  // laço de repetição
   do {
      contador();
      i++;
   } while (i < 1000);
  return(0);
}
```

```
cont:
   .zero
   .globl

contador:
   mov  eax, DWORD PTR cont
   add  eax, 1
   mov  DWORD PTR cont, eax
   ret

main:
   sub  esp, 4
   mov  DWORD PTR [esp], 0
.L3:
   call contador
   add  DWORD PTR [esp], 1
   cmp  DWORD PTR [esp], 999
   jle  .L3
   mov  eax, 0
   add  esp, 4
   ret
```

Figura 3. Código de laço de repetição com chamada interna de procedimento

Nesse cenário, a frequência de execução das instruções de desvio, em relação ao total de instruções executadas, atinge uma relação de 2 para 8. Esse é o caso que mais se aproxima da densidade média de desvios observada em ataques ROP. Mesmo assim, a frequência de saltos ainda é duas vezes menor do que a média de desvios observada em cadeias de *gadgets*. Se a otimização de omissão do ponteiro de *frame* não for empregada, essa densidade cai para 2 desvios a cada 11 instruções.

Na prática, a frequência de desvios quase sempre é menor do que os piores casos apresentados. No caso de códigos compilados, que correspondem à imensa maioria das instruções de máquina executadas, os próprios compiladores eliminam estruturas desnecessárias como aquelas incluídas nos exemplos de códigos analisados (*function inlining optimization*), a fim de otimizar o código gerado. Além disso, a ocorrência de desvios do tipo indireto é pouco comum, independente da situação específica do seu uso [Li et al. 2002]. Ainda assim, caso experimentos futuros identifiquem a ocorrência de falsos positivos durante a análise de executáveis, pode-se verificar se as chamadas ou retornos sequenciais de funções utilizam o mesmo endereço. Essa checagem permite distinguir, mediante um pequeno custo computacional adicional, tanto laços de repetição com chamada a procedimentos pequenos quanto funções recursivas com poucas instruções.

5. Implementação

Para viabilizar o desenvolvimento da proteção através da instrumentação binária dinâmica de código, necessária para a análise em tempo real das instruções executadas, utilizou-se o instrumentador binário Pin. Em seguida, são detalhados os aspectos de implementação da estratégia proposta.

5.1. Pin

O Pin é um *framework* de instrumentação binária dinâmica do tipo JIT (*Just-in-time*), desenvolvido pela Intel para as arquiteturas IA-32 e x86-64, que permite a análise e eventual modificação do código à medida que ele é executado. Para isso, antes que uma instrução seja executada pelo processador, esse *framework* intercepta a instrução, gera e executa novos códigos, e garante que o *framework* retomará o controle do processador após a execução da instrução [Luk et al. 2005]. Como trata-se de uma ferramenta de instrumentação binária dinâmica, a instrumentação é realizada na etapa de execução de arquivos binários previamente compilados. Portanto, o Pin não requer a recompilação de códigos-fontes e permite a análise de programas que geram códigos dinamicamente.

As ferramentas criadas utilizando-se o Pin, chamadas de Pintools, podem ser usadas para a análise de programas pertencentes ao espaço de aplicações do usuário nos sistemas operacionais Android, Linux, OSX e Windows. Cada Pintool possui um mecanismo que decide onde e qual código deve ser inserido, denominado código de instrumentação, e um código a ser enxertado nos pontos de inserção, denominado código de análise [Intel 2014]. É importante ressaltar que o Pin, a Pintool e a aplicação não compartilham nenhuma biblioteca, o que evita qualquer tipo interação não desejada entre esses três binários.

O Pin foi escolhido como base para o protótipo por apresentar o melhor desempenho entre as aplicações de instrumentação binária dinâmica [Luk et al. 2005, Guha et al. 2007], além de fornecer uma API que facilita o acesso a informações de contexto, como o conteúdo de registradores ou o endereço de instruções.

5.2. Módulo de controle da frequência de desvios indiretos

A proteção elaborada requer a criação de uma estrutura de armazenamento, aqui designada 'janela', para registrar as últimas instruções executadas. Assumindo-se que a janela possui um tamanho N, pode-se dizer que sua função é permitir a contagem do número de desvios indiretos executados nas últimas N instruções. Nessa janela, as posições correspondentes às instruções de desvio indireto são anotadas com um bit 1 e as demais instruções são representadas pelo bit 0.

A Figura 4 ilustra a lógica de verificações utilizada para controlar a execução de instruções de desvio indireto. Ao executar qualquer instrução, a janela precisa ser atualizada. Para evitar um *overhead* excessivo decorrente da análise de todas as instruções de um programa, na abordagem de instrumentação binária dinâmica provida pelo Pin é possível explorar o conceito de bloco básico (BBL) [Intel 2014]. Assim, insere-se um código de análise para um BBL, ao invés de avaliar cada instrução do programa, tornando a instrumentação mais eficiente.

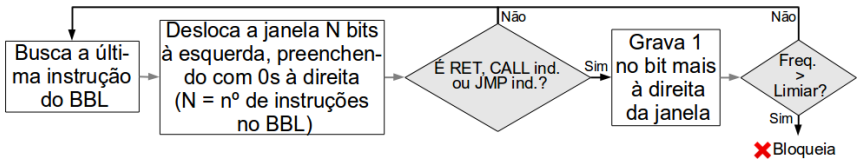


Figura 4. Lógica de controle das instruções de desvio indireto

No esquema proposto, utiliza-se uma API do Pin (BBL_InsTail) para buscar a última instrução do BBL que está sendo instrumentado. Como, por definição, um

BBL é um bloco de instruções com um único ponto de entrada e um único ponto de saída, sabe-se que a única instrução desse bloco que eventualmente poderá corresponder a um desvio indireto será a última instrução do bloco. Depois de capturar a última instrução do BBL, o protótipo desloca a janela de instruções à esquerda, preenchendo os bits deslocados à direita com o bit zero (0). O número de bits deslocados corresponde à quantidade de instruções existentes no BBL em análise. Essa operação de deslocamento da janela obrigatoriamente deve ser realizada para todos os BBLs executados pela aplicação, independente de eles possuírem alguma instrução de desvio indireto ou já terem sido executados anteriormente. Esse é um dos principais fatores que pesa negativamente no desempenho da proteção, já que acarreta em uma mudança de contexto entre o *framework* e a aplicação instrumentada a cada execução de um BBL. Considerando que boa parte dos laços de repetição são mapeados para BBLs, é fácil constatar o impacto dessa característica no desempenho de aplicações que possuem muitos laços de repetição curtos. Infelizmente, em função da forma como o Pin foi concebido, não foi possível evitar esse *overhead* no funcionamento do protótipo.

Após deslocar a janela de instruções, checa-se a última instrução do BBL. Caso ela não corresponda a um desvio indireto, nada mais é preciso ser feito e a execução da aplicação prossegue até que um novo BBL seja buscado. Por outro lado, se a última instrução do BBL corresponder a um desvio indireto, o protótipo grava o valor um (1) no bit mais à direita da janela e calcula a quantidade de desvios indiretos registrados na janela. Caso o valor calculado ultrapasse o limiar estabelecido para a aplicação, o protótipo sinaliza a ocorrência de um ataque ROP e encerra a execução da aplicação.

6. Avaliação e Resultados Experimentais

Esta seção apresenta e discute os resultados dos experimentos realizados. Para tanto, inicialmente o ambiente de experimentação é detalhado. Em seguida, os resultados do processo de validação são mostrados, comprovando a viabilidade dessa solução. Também são discutidos os experimentos realizados para comprovar a eficácia do protótipo. Por fim, uma análise do desempenho do protótipo em comparação com soluções correlatas é apresentada.

6.1. Ambientes de experimentação

Os testes de desempenho foram executados em um computador com processador Pentium E5800 3.20GHz, Dual-Core (cache de 2048 KB), 6GB de memória, sistema operacional Linux Ubuntu 12.04 x86_64 kernel 3.8.0-38-generic, compilador GCC 4.6.3 e Pin versão 2.13 (kit 62728). Para a validação da estratégia de proteção foi usado um computador com processador Intel Xeon E5-2630 2.30GHz, Hexa-Core (cache de 15360 KB), 32 GB de memória, mesmo sistema operacional, compilador e versão do Pin.

6.2. Validação da estratégia de proteção

A estratégia de proteção contra ataques ROP proposta neste trabalho, baseada no controle da frequência de instruções de desvio indireto, foi validada através da análise comparativa entre a frequência de desvios indiretos observada em *exploits* ROP e em aplicações autênticas, representadas pelos *benchmarks* da suíte SPEC CPU2006.

As Figuras 5 (a e b) e 6 (a e b) ilustram, respectivamente, os resultados obtidos nos experimentos com o Linux para janelas de 32, 96, 64 e 128 instruções. Note que na Figura 6, onde são exibidos os gráficos referentes às janelas de 64 e 128 instruções, cada *benchmark* possui duas marcações. Isso acontece porque, nesses casos, cada *benchmark* foi compilado para duas arquiteturas: 32 bits e 64 bits.

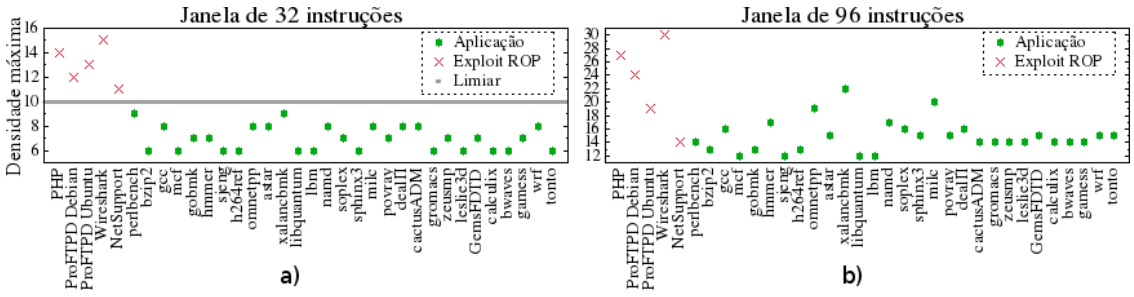


Figura 5. Frequência máxima de instruções de desvio indireto registrada com janelas de 32 e 96 instruções no Linux

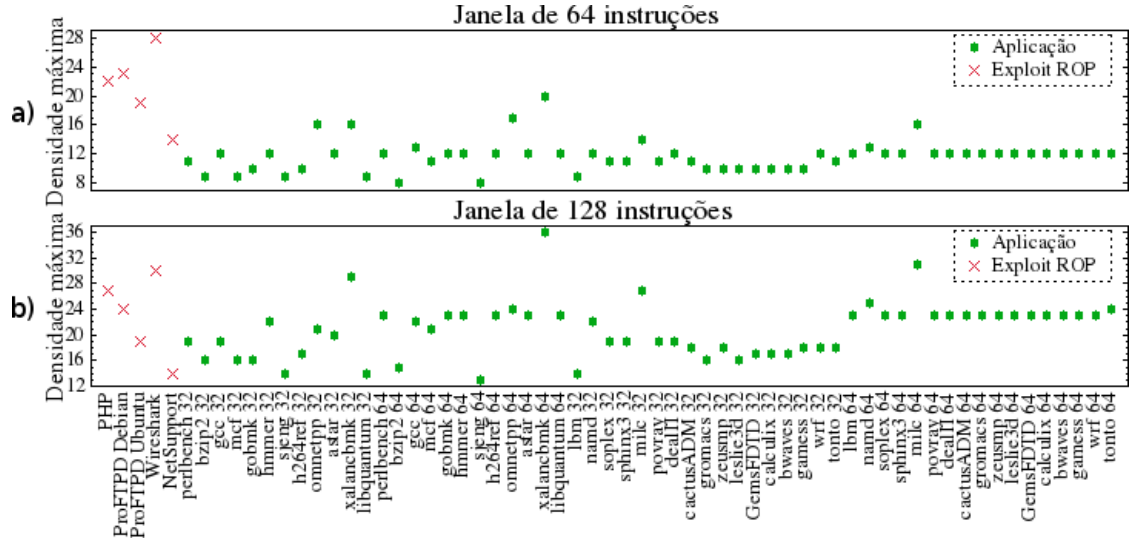


Figura 6. Frequência máxima de instruções de desvio indireto registrada com janelas de 64 e 128 instruções no Linux

Ao analisar as Figuras 5 e 6, é possível constatar que a densidade máxima de instruções de desvio indireto tende a ser maior nos *exploits* ROP do que em aplicações autênticas. No entanto, quanto maior o tamanho da janela, menor é a discrepância dos resultados entre as duas classes de executáveis analisadas. Tanto que, no caso da janela de 128 instruções, cujos resultados são apresentados na Figura 6.b, não é possível distinguir os *exploits* das aplicações. Em contrapartida, à medida que o tamanho da janela diminui para 96 (Figura 5.b), 64 (Figura 6.a) ou 32 (Figura 5.a) instruções, essa distinção torna-se nítida. Apesar disso, o único tamanho de janela testado para o qual foi possível estabelecer um limiar universal, teoricamente capaz de distinguir qualquer aplicação de qualquer *exploit* ROP, foi o de 32 instruções. Em outras palavras, para a janela de 32 instruções, pode-se

traçar uma linha (conforme indicado na Figura 5.a) que separa os dois padrões de frequência máxima de desvios indiretos, já que os valores registrados para *exploits* ROP variaram de 11 a 15, enquanto nos *benchmark* esses valores se espalharam entre 6 e 9. Portanto, pelos resultados obtidos, um limiar de 10 desvios indiretos é capaz de distinguir aplicações autênticas de *exploits* ROP, quando empregado junto com uma janela de 32 instruções.

Apesar de os resultados indicarem a possibilidade de se estabelecer um valor padrão para separar a frequência máxima de desvios indiretos apresentada por um *exploit* ROP daquela atingida por aplicações autênticas, a definição de um limiar universal pode não ser totalmente confiável, em função da proximidade entre as fronteiras. Uma solução mais robusta é usar como limiar a frequência máxima de desvios indiretos específica de cada aplicação, porque isso permite a redução do limiar para a maioria das aplicações e, consequentemente, amplia a diferença em relação aos *exploits* ROP. Para ficar mais claro como essa abordagem pode beneficiar um mecanismo de proteção contra ataques ROP baseado no controle da frequência de desvios indiretos, basta observar-se que poucas aplicações apresentam uma frequência máxima de desvios indiretos próxima do limiar geral.

A distribuição estatística das frequências máximas de desvios indiretos entre os *benchmarks* monitorados com a janela de 32 instruções mostra que apenas 6,9% dos experimentos alcançaram uma frequência máxima de desvios indiretos igual a 9 (nove), que corresponde ao valor mais próximo do limiar universal (10). Além disso, a frequência máxima de desvios indiretos mais comum é justamente a mais distante do limiar universal (6), abarcando 41,4% dos casos. As frequências 8 e 7 representam, respectivamente, 27,6% e 24,1% dos casos. Essa distribuição estatística permite inferir que, ao ampliar-se o escopo de experimentos para uma gama ainda maior de aplicações, a tendência é de que poucos casos se aproximem do limiar universal.

6.3. Eficácia no bloqueio de *exploits* reais

O protótipo foi testado na proteção de 5 aplicações para as quais existem *exploits* ROP publicamente disponíveis no repositório Exploit Database. Foram utilizados os mesmos exemplares de códigos maliciosos empregados na validação da estratégia de proteção. Os experimentos foram executados em duas etapas. Na primeira, o correto funcionamento dos *exploits* foi confirmado através da reprodução dos ataques contra máquinas virtuais onde as aplicações vulneráveis foram instaladas. Na sequência, executou-se essas aplicações sob o controle do protótipo e repetiu-se os ataques. Dessa forma, pôde-se observar a eficácia da estratégia de controle da frequência de desvios indiretos na proteção contra ataques ROP reais. Em todos os casos, o protótipo foi capaz de detectar o ataque ROP e impedir a sua consolidação.

6.4. Desempenho

Os experimentos realizados para avaliar o desempenho do protótipo desenvolvido estão resumidos na Tabela 1. Nela, estão expressos os *overheads* impostos pelo Pin e pelo protótipo ao executar todos os *benchmarks* da suíte SPEC CPU2006 nas arquiteturas de 32 bits e de 64 bits. O Pin foi executado sem a adição de qualquer Pintool, com o intuito de registrar o custo computacional mínimo imposto pelo *framework*. Seu *overhead* médio atingiu 29%. Esse resultado confirma a expectativa de

um significativo custo computacional imposto por instrumentadores binários dinâmicos [Luk et al. 2005]. Da mesma forma, o fato do Pin acarretar em um *overhead* maior com os *benchmarks* inteiros confirma os resultados reportados pelos autores dessa ferramenta.

Tabela 1. *Overhead* médio ao executar *benchmarks*

	***Overhead* (%)**	
Benchmark	**Pin**	**Protótipo**
SPEC CPU2006 FP 32	10	1278
SPEC CPU2006 INT 32	56	801
SPEC CPU2006 FP 64	13	730
SPEC CPU2006 INT 64	37	756
Média Geral	**29**	**891**

Por outro lado, na arquitetura de 32 bits, o protótipo apresentou um *overhead* maior quando executou os *benchmarks* de ponto flutuante. Mesmo na arquitetura de 64 bits, os resultados obtidos para os testes inteiros e de ponto flutuante foram próximos. Esse distanciamento em relação ao padrão apresentado pelo Pin ocorreu porque o principal fator de influência no desempenho do protótipo é a quantidade de BBLs executados. Isso acontece porque a estratégia de controle da frequência de desvios indiretos exige que uma função de análise seja lançada ao executar qualquer BBL. Se observarmos que cada repetição de um laço corresponde a um BBL no Pin, fica fácil entender porque o desempenho do protótipo tende a ser pior com os *benchmarks* do tipo ponto flutuante, que executam inúmeras repetições de laços.

Outra constatação decorrente dos resultados reportados na Tabela 1 reside no fato de que para a arquitetura de 64 bits, tanto o Pin quanto o protótipo apresentaram um desempenho melhor. Isso ocorre em função das otimizações incorporadas ao processador utilizado nos experimentos, que beneficiam códigos de 64 bits.

A Tabela 2 exibe uma comparação do protótipo desenvolvido neste trabalho com outras proteções contra ataques ROP que utilizam instrumentadores binários dinâmicos para implementar soluções baseadas na estratégia de controle das instruções de retorno. Nessa tabela estão expressos os tipos de ataques bloqueados pelas proteções e o *overhead* médio reportado pelos autores de cada solução. Isso significa que os resultados remontam a diferentes conjuntos de teste e ambientes de experimentação. Apesar disso, pode-se dizer que o protótipo desenvolvido apresenta um custo computacional comparável à solução DROP [Chen et al. 2009], que também utiliza o Pin, mas mediu o desempenho através da execução de uma seleção de aplicações, ao invés da suíte de *benchmarks* SPEC. As únicas aplicações utilizadas tanto nos testes executados com o DROP quanto nos experimentos realizados com o nosso protótipo (bzip2 e gcc), que podem oferecer uma comparação um pouco mais realista, indicam que o protótipo desenvolvido neste trabalho impõe um *overhead* menor. Nos experimentos com o bzip2, o DROP acarretou em um custo computacional de 1.540%, consideravelmente superior aos 721% registrado pelo nosso protótipo. Nos testes com o gcc, o DROP impôs um *overhead* de 960%, enquanto o nosso protótipo elevou o tempo de CPU em 830%.

Outro fator de comparação entre as proteções recai sobre os tipos de ataques ROP bloqueados. Nesse caso, conforme indicado na Tabela 2, apenas a solução desenvolvida neste trabalho oferece uma proteção contra todos os tipos de *exploits*

Tabela 2. Comparação das proteções contra ataques ROP

Proteção	Ataques Bloqueados	*Overhead* (%)
DynIMA	R	Não informado na publicação
DROP	R	530,0
Protótipo	**R, J e C**	**891,0**

Ataques bloqueados: R-encadeamento via RET; J-encadeamento via JMP; C-encadeamento via CALL.

ROP. Essa capacidade está diretamente relacionada à mudança na estratégia de detecção dos ataques adotada neste projeto, que amplia o escopo de monitoramento para abarcar todas as instruções de desvio indireto.

7. Conclusões e Trabalhos Futuros

Este trabalho demonstrou que a imposição de um limite para o uso de instruções de desvio indireto acarreta em severas limitações à capacidade de criação de um *exploit* ROP efetivo, impossibilitando-a em todos os casos testados. Além disso, a estratégia de controle da frequência de instruções de desvio indireto possibilita o bloqueio das demais variantes de ataques ROP, fato inédito entre as soluções que utilizam uma estratégia de controle da frequência de instruções. Finalmente, foi desenvolvido neste trabalho um protótipo que pode ser facilmente adotado em ambientes de produção que executem os sistemas operacionais Linux, Windows, Android ou OSX. A análise de desempenho do protótipo indicou que as contribuições são obtidas a um custo computacional comparável à de soluções correlatas, superando-as em alguns casos.

Entre os projetos futuros que podem dar prosseguimento a este trabalho, a avaliação do protótipo em outros ambientes já está em andamento. Além disso, está sendo elaborada uma solução para tratar eventuais casos de falsos positivos decorrentes de funções com recursividade em cauda. Outra possibilidade de trabalho futuro consiste em utilizar abordagens alternativas para implementar a estratégia de controle da frequência de desvios indiretos que permitam reduzir o *overhead* computacional. Entre elas, uma opção é adaptar estruturas de hardware disponíveis nos processadores atuais e originalmente desenvolvidas para outras finalidades, como o *Return Stack Buffer* (RSB), o *Last Branch Recording* (LBR) e o *Branch Trace Store* (BTS), com o intuito de implementar a estratégia descrita neste artigo.

Referências

Bania, P. (2010). Security mitigations for return-oriented programming attacks. *CoRR*, abs/1008.4099.

Checkoway, S., Davi, L., Dmitrienko, A., Sadeghi, A.-R., Shacham, H., and Winandy, M. (2010). Return-oriented programming without returns. In *Proceedings of the 17th ACM CCS*, pages 559–572. ACM.

Checkoway, S., Feldman, A. J., Kantor, B., Halderman, J. A., Felten, E. W., and Shacham, H. (2009). Can dres provide long-lasting security? In *Proceedings of the EVT/WOTE*, pages 6–6. USENIX Association.

Chen, P., Xiao, H., Shen, X., Yin, X., Mao, B., and Xie, L. (2009). Drop: Detecting return-oriented programming malicious code. In *Information Systems Security*, pages 163–177. Springer Berlin Heidelberg.

Chen, P., Xing, X., Mao, B., Xie, L., Shen, X., and Yin, X. (2011). Automatic construction of jump-oriented programming shellcode (on the x86). In *Proceedings of the ACM ASIACCS*, pages 20–29. ACM.

D. Rosenberg (2011). Defeating windows 8 rop mitigation. http://goo.gl/2Ae7aN.

Davi, L., Sadeghi, A.-R., and Winandy, M. (2009). Dynamic integrity measurement and attestation. In *Proceedings of the ACM STC*, pages 49–54. ACM.

Ferreira, M. T., Rocha, T., Martins, G., Feitosa, E., and Souto, E. (2012). Analise de vulnerabilidades em sistemas computacionais modernos: Conceitos, exploits e protecoes. In *Livro de Minicursos do XII SBSeg*, pages 2–51. SBC.

Guha, A., Hiser, J. D., Kumar, N., Yang, J., Zhao, M., Zhou, S., Childers, B. R., Davidson, J. W., Hazelwood, K., and Soffa, M. L. (2007). Virtual execution environments: Support and tools. In *NSF Next Generation Software Program Workshop*, Long Beach, CA.

Hoglund, G. and McGraw, G. (2004). *Exploiting Software: How to Break Code*. Pearson Higher Education.

Intel (2014). Pin 2.13 user guide. http://goo.gl/xvsW6l.

J. Callas (2011). Smelling a rat on duqu. http://goo.gl/FTM1Jn.

J. McDonald (1999). Defeating solaris/sparc non-executable stack protection. http://goo.gl/fglJTX.

Jiang, J., Jia, X., Feng, D., Zhang, S., and Liu, P. (2011). Hypercrop: A hypervisor-based countermeasure for return oriented programming. In *Proceedings of the 13th ICICS*, pages 360–373, Berlin, Heidelberg. Springer-Verlag.

Li, T., Bhargava, R., and John, L. K. (2002). Rehashable btb: An adaptive branch target buffer to improve the target predictability of java code. In *In The International Conference on High Performance Computing (HiPCP*.

Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., Wallace, S., Reddi, V. J., and Hazelwood, K. (2005). Pin: Building customized program analysis tools with dynamic instrumentation. In *Proceedings of the ACM SIGPLAN PLDI*, pages 190–200. ACM.

Min, J.-W., Jung, S.-M., and Chung, T.-M. (2013). Detecting return oriented programming by examining positions of saved return addresses. In *Ubiquitous Information Technologies and Applications*, pages 791–798. Springer Netherlands.

N. H. Son (2011). Rop chain for windows 8. http://goo.gl/MAujbX.

S. Designer (1997). Getting around non-executable stack (and fix). http://goo.gl/XNEE7n.

S. Krahmer (2005). x86-64 buffer overflow exploits and the borrowed code chunks exploitation technique. http://goo.gl/5cN0Bm.

Shacham, H. (2007). The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86). In *Proceedings of the 14th ACM CCS*, pages 552–561. ACM.

T. Newsham (1997). Re: Smashing the stack: prevention? http://goo.gl/fglJTX.

Universitet, S. (2014). Intel and att syntax. http://goo.gl/gkTvxL.

Wojtczuk, R. N. (2001). The advanced return-into-lib(c) exploits: PaX case study. *Phrack*, 11(58).

Yuan, L., Xing, W., Chen, H., and Zang, B. (2011). Security breaches as pmu deviation: Detecting and identifying security attacks using performance counters. In *Proceedings of the Second APSys*, pages 6:1–6:5. ACM.

Estruturas Virtuais e Diferenciação de Vértices em Grafos de Dependência para Detecção de *Malware* Metamórfico.

Gilbert B. Martins, Eduardo Souto, Rosiane de Freitas, Eduardo Feitosa

Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

{gilbert.martins, esouto, rosiane, efeitosa}@icomp.ufam.edu.br

Abstract. *This paper presents a methodology for identifying metamorphic malware based on the comparison of dependency graphs stored in a reference. On the strength of the structural differentiation of the vertices and the addition of virtual structures, the proposed methodology is able to identify and eliminate non-relevant elements of the original reference graph, reducing the size of the reference database and improving the results obtained during the comparison of the graphs. To validate this, is presented the comparison of results generated by the proposed approach with those from a reference method in the identification of W32.Evol and W32.Polip metamorphic malwares.*

Resumo. Este artigo apresenta uma metodologia de identificação de *malware* metamórfico baseada na comparação de grafos de dependência armazenados numa base de referência. Em função da diferenciação estrutural dos vértices e da adição de estruturas virtuais, a metodologia proposta é capaz de identificar e eliminar os elementos não relevantes do grafo de referência original, reduzindo o tamanho da base de referência e melhorando a variância nos resultados obtidos durante a comparação entre os grafos. Para validar isto, é apresentada a comparação dos resultados obtidos com aqueles gerados por uma metodologia de referência, na identificação dos *malware* metamórficos W32.Evol e W32.Polip.

1. Introdução

A última década tem enfrentado um aumento significativo no número e na sofisticação dos ataques digitais baseados em *malwares*[1] [Baker et al. 2011]. Fatores como o uso da Internet para a distribuição massificada e a capacidade de auto-propagação por redes locais, aumentam consideravelmente o grau de dificuldade do combate a estas ameaças. Para tratar este problema, é comum fazer uso de uma base de dados contendo trechos de código extraídos de cada *malware*, que associada a um processo de varredura de programas suspeitos, permite a identificação de códigos maliciosos [Karin 2006]. É importante salientar que, para o processo de identificação ter sucesso, o trecho de código

[1] O termo *malware* (do inglês, *malicious software*) é usado para classificar um software destinado a se infiltrar em um sistema de computador alheio de forma ilícita, com o intuito de causar algum dano ou roubo de informações (confidenciais ou não).

selecionado como assinatura deve ser único o suficiente para ser encontrado apenas no *malware* que se pretende identificar [Moura e Rebiha 2009].

Apesar de eficiente, a identificação pelo uso de assinaturas só classificará como *malware*, programas que possuírem, dentro de sua codificação, um trecho de código que seja idêntico a uma das assinaturas catalogadas. Tal premissa tem sido explorada pelos desenvolvedores de códigos maliciosos, que empregam técnicas de ofuscação de código para dificultar o processo de detecção [Borello e Mé 2008][Notoatmodjo 2010]. Tais técnicas têm como objetivo mascarar a identidade do programa malicioso e se baseiam na modificação da sequência original de instruções sem prejuízo à funcionalidade original dos trechos alterados. Exemplos de técnicas comuns de ofuscação incluem [Bruschi et al. 2007]: *i)* a inserção de instruções e variáveis irrelevantes, também conhecida como inserção de código lixo, que não alteram a lógica original do programa; *ii)* a alteração no nome de variáveis ou troca mútua de variáveis entre instruções diferentes; *iii)* a substituição de sequências de instruções por outras que produzam o mesmo resultado; e *iv)* a alteração na ordem de execução das instruções, seja pelo reposicionamento de blocos de código independentes ou pelo uso de instruções de desvio de fluxo.

Essa capacidade de mutação do código original também é conhecida como metamorfismo de código [Borello e Mé 2008]. As versões metamórficas de um *malware* são geradas automaticamente por um componente do código (*engine* de metamorfismo) que é incorporado no próprio *malware* e tem a função de executar as alterações no código à medida que novas cópias do *malware* são produzidas e propagadas. Assim, mesmo pequenas alterações no código malicioso podem conduzir a falhas no processo de detecção, o que requer constantes atualizações nas bases de assinaturas. Como o número de versões metamórficas pode crescer exponencialmente, torna-se praticamente impossível sua detecção com base no modelo tradicional de assinaturas.

Diversas abordagens têm sido propostas para lidar com este problema como: a criação de um padrão de assinatura capaz de identificar grupos de códigos através de uma única sequência de identificação [Griffin et al. 2009]; a utilização de autômatos finitos para modelar chamadas de sistemas associadas ao comportamento de códigos maliciosos [Jacob et al. 2009]; a normalização do código e o levantamento do fluxo para reversão e identificação de códigos suspeitos [Cozzolino et al. 2012]; e a utilização de grafos para modelar o uso de funções [Hu et al. 2009] ou a relação de dependência entre instruções do código [Kim e Moon 2010]. Alguns dos problemas enfrentados por estas abordagens são: a criação manual dos modelos de detecção, a quantidade de informações tratadas ou ainda a alta variância nos resultados de identificação.

Este trabalho propõe uma nova metodologia para identificação de *malwares* metamórficos baseada na análise de grafos de dependência, gerados automaticamente a partir da análise de programas executáveis. A metodologia proposta é capaz de identificar as partes relevantes deste grafo, baseando-se nas características estruturais de seus vértices e arestas. Isto permite a criação de um processo mais eficiente de redução de grafos, o que diminui a quantidade de informação necessária para identificar um *malware* metamórfico. Avaliações feitas com coleções de dados reais obtidas a partir de

versões metamórficas do vírus W32.Evol[2] e W32.Polip[3] demonstram melhorias na detecção de códigos metamórficos e diminuição na variância dos resultados quando comparados com a abordagem proposta por Kim e Moon [Kim e Moon 2010], que foi usada como modelo de referência neste texto.

O restante deste artigo está organizado da seguinte forma. A seção 2 fornece alguns trabalhos relacionados ao problema de detecção de *malware* metamórfico. A seção 3 define grafos de dependência e como eles podem ser aplicados para identificar as semelhanças de códigos. A seção 4 detalha a metodologia de identificação proposta. A seção 5 fornece resultados experimentais e discussões. Finalmente, a seção 6 apresenta as conclusões e dá indicações para trabalhos futuros.

2. Trabalhos Relacionados

Abordagens alternativas para o modelo tradicional de assinaturas procuram se basear em modelos de identificação que sejam mais resistentes às técnicas de ofuscação de código empregadas por desenvolvedores de *malware* metamórficos.

Uma das propostas usa um procedimento automatizado para a análise de grupos de *malware* previamente identificados, criando um conjunto mínimo não linear de sequências de *bytes* de tamanho *n*, conhecidas como "*assinaturas string*", baseado na probabilidade de um símbolo vir após uma sequência qualquer de símbolos [Griffin et al. 2009]. Segundo os autores, seria possível melhorar o procedimento pela geração de assinaturas candidatas baseada em múltiplos trechos não consecutivos de código, mas a sobrecarga computacional derivada disto não seria irrelevante.

A utilização de autômatos finitos também pode ser empregada para identificar o comportamento apresentado por códigos maliciosos [Jacob et al. 2009]. Um autômato finito é usado para modelar uma sequência de chamadas a funções do sistema operacional. Em seguida, este autômato deve ser comparado com outros, previamente construídos, que modelam o comportamento apresentado por um *malware*. Entretanto, a necessidade da criação manual destes autômatos requer um grande conhecimento a respeito do fluxo de dados e do comportamento apresentado tanto por códigos maliciosos como por programas benignos, o que não é algo tão simples de se obter.

Outra alternativa combina a normalização e o mapeamento do fluxo de execução dos programas analisados [Cozzolino et al. 2012]. Primeiramente, o processo de normalização deve restaurar o código o mais próximo possível ao seu estado original. Em seguida, este código é dividido em blocos funcionais, delimitados por instruções de desvio de fluxo, e reduzidos a marcadores inteiros simples. Cada marcador é combinado com aqueles correspondentes aos dois possíveis destinos a partir deste bloco, formando uma identificação composta. Ao final, este conjunto de marcadores compostos é comparado com aqueles previamente armazenados em uma base de dados para a geração de uma pontuação que será utilizada para determinar se o código analisado se trata ou não de um *malware*. A principal limitação desta metodologia está associada ao processo de normalização, pois se este processo não obtiver os resultados esperados a

[2] http://www.symantec.com/ security_response/writeup.jsp?docid=2000-122010-0045-99

[3] http://www.symantec.com/ security_response/writeup.jsp?docid=2006-042309-1842-99

pontuação final gerada no processo de comparação será muito baixa, impedindo a correta identificação do *malware.*

Uma metodologia baseada em grafos [Hu et al. 2009] propõe a análise de códigos executáveis para a construção de uma estrutura que modele as chamadas de função presentes no código. No grafo gerado a partir desta análise, cada vértice v_i está associado a uma função e uma aresta $v_a v_b$ é criada sempre que no corpo da função v_a existir uma chamada para a função v_b. Este grafo é comparado com uma base de grafos previamente existente, permitindo que a identificação do *malware* ocorra. A maior limitação deste processo está associada à dificuldade de mapear corretamente as funções criadas diretamente no código do programa, tarefa esta que pode ser ainda mais dificultada dependendo da quantidade de técnicas metamórficas aplicadas a este código.

Outro exemplo da utilização de grafos é uma proposta para detecção de códigos maliciosos inseridos dentro de *scripts* [Kim e Moon 2010]. O código suspeito é analisado para a geração de um grafo de dependência que modela as inter-relações entre cada instrução presente no código, baseadas nas variáveis que manipulam. Este grafo passa por um processo de normalização que visa eliminar instruções inseridas pelas ações das técnicas de ofuscação de código, além de diminuir o tamanho do grafo original. O processo de detecção é baseado no problema de encontrar o máximo isomorfismo de subgrafo entre o grafo normalizado e um grafo que modela um código malicioso previamente identificado. Entretanto, como se trata de um problema NP-Difícil, o custo de execução é bem alto, o que exige a utilização de heurísticas para a diminuição do tempo de processamento.

3. Grafos de Dependência na Identificação de Códigos Metamórficos

Grafos de dependência são grafos direcionados que representam relações de dependência entre elementos pertencentes a uma mesma estrutura [Ferrante et al. 1987]. Originalmente empregados na identificação de plágios [Liu et al. 2006], os grafos de dependência também tem sido usado como estrutura base para eliminar as ações das técnicas de ofuscação em códigos maliciosos [Kim e Moon 2010]. Nessa abordagem, cada instrução do código corresponde a um vértice e, para cada variável que esta instrução manipular, uma aresta orientada será inserida, ligando esta instrução à próxima linha de código que manipular esta mesma variável. Este trabalho propõe o uso de grafos de dependência aplicada a códigos executáveis. A Figura 1 apresenta um exemplo de grafo de dependência gerado a partir de um código *assembly*. Este procedimento é aplicado tanto a um programa suspeito de contaminação como no *malware* que será investigado, gerando dois grafos que serão então reduzidos, para eliminação de componentes derivados do metamorfismo (Figura 1.c), e finalmente comparados, para identificação de similaridades. Encontrar correspondências entre os vértices de dois grafos dados recai no problema conhecido como *Isomorfismo entre Grafos* [Garey e Johnson 1979].

Baseado nestes conceitos, este trabalho utiliza uma metodologia de identificação de códigos executáveis metamórficos de origem maliciosa, através da conversão destes códigos para linguagem *assembly* e posterior construção dos grafos de dependência correspondentes, que serão então comparados com uma base de referência para sua identificação como *malware*.

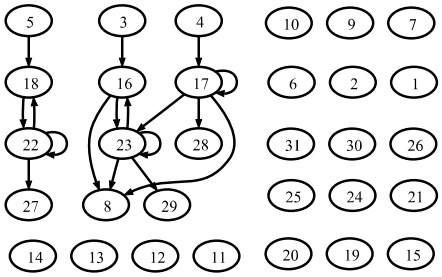
4. Metodologia de Identificação Utilizada

O processo de identificação de códigos executáveis metamórficos é composto por quatro etapas principais:

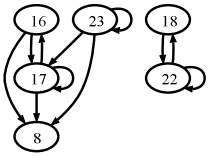
1. Reconstrução de código *assembly*, onde o programa executável passa por um processo de engenharia reversa para a obtenção do seu código equivalente em linguagem *assembly*. Esta etapa pode ser executada com o auxílio de programas como OllyDbg [4] e IDA Pro [5].
2. Geração do grafo de dependência, onde o programa gerado na etapa anterior é analisado e então usado como base para a geração do grafo de dependência.
3. Redução do grafo, usado para reduzir o grafo de dependência obtido na etapa anterior. Partes do código onde o controle de fluxo nunca irá passar são removidas. De acordo com nossa proposta, um tratamento adicional de redução deve ser executado para os grafos que constituírem a base de referência.
4. Comparação do grafo reduzido com a base de referência, onde o grafo reduzido é comparado com um grafo correspondente a um *malware* previamente analisado.

```
linha:01 .686p
linha:02 .model flat
linha:03    push eax
linha:04    push ebx
linha:05    push ecx
linha:06    call sub_01
linha:07 ini_loop:
linha:08    cmp ebx, eax
linha:09    jg  end_loop
linha:10    call sub_02
linha:11    jmp ini_loop
linha:12 end_loop:
linha:13    call sub_03
linha:14    end
linha:15 sub_01 proc near
linha:16    mov eax, 9
linha:17    mov ebx, 3
linha:18    mov ecx, 0
linha:19 retn
linha:20 sub_01 endp
linha:21 sub_02 proc near
linha:22     add ecx, 1
linha:23     sub eax, ebx
linha:24 retn
linha:25 sub_02 endp
linha:26 sub_03 proc near
linha:27    pop ecx
linha:28    pop ebx
linha:29    pop eax
linha:30 retn
linha:31 sub_03 endp
```

a) Código assembly



b) Grafo de dependência original



c) Grafo de dependência reduzido

Figura 1. Exemplo de um código *assembly* (a) e seus grafos de dependência original (b) e reduzido (c), construídos a partir das dependências do código semântico.

4.1. Geração dos Grafos de Dependência

O código *assembly* gerado na primeira etapa da metodologia é submetido a um processo que mapeia as relações de dependência entre instruções, registradores e variáveis que cada instrução manipula. Cada instrução é associada com um vértice e a manipulação dos registradores/variáveis definirá quais arestas serão criadas.

[4] http://www.ollydbg.de.

[5] http://www.hex-rays.com/products/ida/index.shtml.

O processo de geração das arestas inicia com a identificação dos registradores/variáveis manipulados em cada instrução. Para cada um desses elementos de armazenamento de dados, uma das seguintes ações pode ser tomada: *a)* caso o elemento esteja sendo manipulado pela primeira vez, o vértice correspondente àquela instrução é marcado como origem para futuras manipulações daquele mesmo elemento; *b)* caso o elemento já tenha uma origem definida, é criada uma nova aresta direcionada no grafo partindo da origem e tendo como destino o vértice correspondente à instrução atual, e; *c)* se a instrução estiver alterando o conteúdo do elemento, além da criação de uma nova aresta, a origem é atualizada para o vértice correspondente à instrução atual.

Este processo ainda prevê a necessidade de reavaliação de instruções, em função da presença de instruções de desvio de fluxo do programa. Isto ocorre porque a origem pode ter sido alterada dentro de um laço ou chamada de procedimento, o que cria novas relações de dependência que devem ser analisadas. Entretanto, este processo deve ser executado com algum cuidado, principalmente no caso das instruções de desvio condicional, visto que a linguagem *assembly* não possui instruções de controle de fluxo (por exemplo, o `if-then-else`) ou instruções de laço, uma vez que todos os controles são implementados através de instruções de desvio do tipo *"jump"*. As relações de dependência mapeadas devem ser compatíveis com o fato de que:

a) Trechos de código podem ser ignorados, o que abre a possibilidade do estabelecimento de relações de dependência entre as instruções localizadas antes e depois do trecho ignorado (`if-then-else`);

b) Trechos de código podem ser executados mais de uma vez, o que pode gerar relações de dependência entre instruções posicionadas em porções de código anteriores ao da instrução atual, além de relações de dependência de uma instrução para ela mesma (um laço).

Assim, para cada desvio condicional presente, abrem-se dois possíveis caminhos alternativos para o fluxo de execução. Os caminhos alternativos podem então ser modelados como uma árvore binária com 2^n folhas, onde n representa a quantidade de instruções de desvio condicional presentes no código.

4.2. Redução dos Grafos de Dependência

Reduzir um grafo de dependência significa eliminar os vértices que são considerados desnecessários, tais como vértices que representam a declaração de variáveis ou que representam trechos do código que nunca serão atingidos. Ao fim do processo de redução obtém-se um grafo que representa o comportamento principal do código, como o exemplo mostrado na Figura 1.c, que representa o resultado da redução aplicada ao grafo mostrado na Figura 1.b.

Para realizar a redução dos grafos de dependência foram definidas quatro situações onde os vértices devem ser eliminados [Kim e Moon 2010]: 1) vértices com apenas uma aresta de saída e sem arestas de entrada; 2) vértices com apenas uma aresta de entrada e sem arestas de saída; 3) vértices com apenas uma aresta de entrada e uma aresta de saída; e 4) vértices que não possuem nenhuma aresta de entrada ou saída.

4.3 Identificando Elementos Relevantes do Grafo

Na metodologia proposta por Kim e Moon [Kim e Moon 2010], os grafos de dependência, gerados com base em códigos maliciosos metamórficos previamente identificados, são armazenados após o processo de redução e utilizados diretamente como base de referência para o processo de identificação de outras versões metamórficas destes mesmos *malware*, sem que qualquer informação a respeito da natureza e função dos vértices seja levada em consideração durante todas as etapas restantes da metodologia. Na proposta apresentada neste artigo, um tipo específico de vértice, denominado de *vértice de decisão*, derivado das operações de comparação entre o conteúdo de registradores, é tratado para se chegar a uma versão ainda mais reduzida do grafo de dependência.

Os vértices de decisão são gerados a partir de instruções `CMP` que são executadas antes de qualquer instrução de *jump* condicional. Como programas em linguagem *assembly* não possuem estruturas de controle de alto nível, como `if-then-else` e `do-while`, esta instrução é a principal ferramenta para a implementação das estruturas de decisão e controle de fluxo do programa. Como estas instruções não alteram o conteúdo dos registradores, os vértices gerados a partir das mesmas possuem a característica singular de não possuírem arestas que se originam a partir destes vértices.

A Figura 2, gerada a partir de um trecho de código do vírus W32.Evol, ilustra um grafo de dependência onde os vértices de decisão estão em destaque. Como as arestas representam as relações de dependência entre as instruções, quanto maior for a quantidade de arestas incidindo neste tipo de vértice, maior será sua importância para a implementação da lógica básica do programa modelado. É comum também surgir mais de um componente conexo nestes grafos, derivados de elementos de controle que não são relevantes para o processo de identificação. Por exemplo, na Figura 1.c, o componente formado pelas arestas 18 e 22 é derivado da manipulação de um contador, podendo ser desconsideradas no processo de identificação. A Figura 3 ilustra um grafo de dependência, gerado a partir de um *malware* real, onde podem ser identificados mais de um componente conexo.

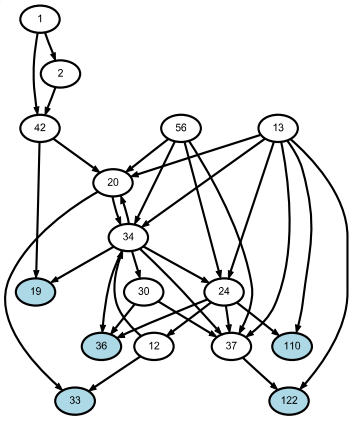


Figura 2. Grafo de dependência reduzido, com os vértices de decisão em destaque.

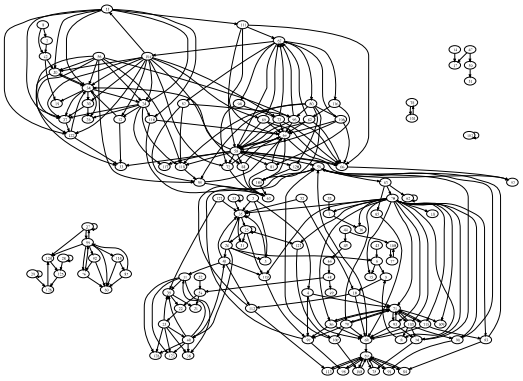


Figura 3. Grafo de dependência do *malware* W32.Evol.

Para identificar os vértices de decisão mais relevantes para o programa modelado, foi utilizado um processo dividido em quatro etapas: 1) cálculo da menor distância relativa entre cada vértice de decisão; 2) construção de um grafo virtual

derivado, constituído apenas dos vértices de decisão, com arestas representando a distância relativa entre cada vértice de decisão; 3) cálculo da clique máxima [Bomze et al. 1999] presente neste grafo virtual derivado; e 4) redução final do grafo de dependência, com a eliminação de qualquer vértice e aresta que não estejam associados aos vértices de decisão presentes na clique máxima do grafo virtual derivado.

4.3.1 Cálculo da Menor Distância Relativa entre Vértices de Decisão

Tradicionalmente, a utilização de um algoritmo como Floyd-Warshall [Floyd 1962] [Warshall 1962] poderia ser empregada para a obtenção da distância entre cada vértice de um grafo. Entretanto, como os vértices de decisão não possuem arestas que se originam a partir deles, a distância calculada a partir destes vértices para qualquer outro vértice presente no grafo de dependência seria sempre infinita.

Assim, no momento do cálculo da distância entre cada vértice de decisão, esta metodologia considera a existência do conjunto de arestas virtuais que invertem o sentido das arestas originalmente incidentes nos vértices de decisão, criando uma conexão de saída que permite o cálculo da distância relativa entre estes vértices. A Figura 4 ilustra as arestas virtuais (destacadas em vermelho) criadas para o grafo apresentado na Figura 2.

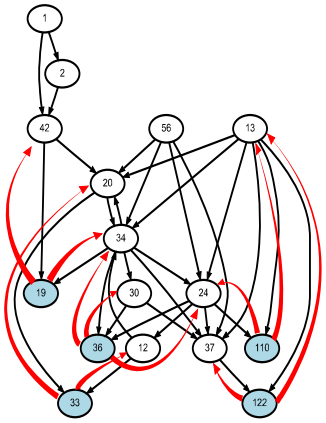


Figura 4. Arestas virtuais adicionadas ao grafo de dependência da Figura 2.

Para que estas arestas virtuais não interfiram no cálculo das distâncias mínimas para os outros nós, foi desenvolvida uma adaptação do algoritmo de Floyd-Warshall observando dois pontos fundamentais: 1) todos os vértices de decisão não são considerados como elementos intermediários para o cálculo das distâncias mínimas entre os vértices; e 2) quando os vértices de origem são identificados como "de decisão" as arestas virtuais são consideradas no cálculo das distâncias mínimas entre este vértice e todos os demais. O algoritmo modificado é apresentado no Algoritmo 1.

Algoritmo 1 Floyd-Warshall Modificado

```
1: for each v_k in G
2:   if v_k isn't a decision vertex
3:     for each v_o
4:       for each v_d
5:         if v_o isn't a decision vertex
6:           if (d(v_o,v_k) + d(v_k,v_d)) < d(v_o,v_d)
8:             set d(v_o,v_d)=(d(v_o,v_k) + d(v_k,v_d))
9:         else
10:          if (d(v_k,v_o) + d(v_k,v_d)) < d(v_o,v_d)
11:            set d(v_o,v_d)=(d(v_k,v_o) + d(v_k,v_d))
```


4.3.2 Construção do Grafo Virtual Derivado

Com todas as distâncias mínimas calculadas, a etapa seguinte cria um grafo virtual onde cada vértice corresponderá a um dos vértices de decisão do grafo de dependência original e as arestas serão geradas com base na distância calculada entre cada um destes vértices. Caso este grafo virtual fosse gerado com base no grafo apresentado na Figura 4, a matriz de adjacências apresentada na Figura 5 seria produzida. É interessante destacar que as distâncias calculadas entre dois vértices nem sempre são as mesmas, dependendo do sentido da aresta.

	19	**33**	**36**	**110**	**122**
19	2	3	2	3	3
33	3	2	3	4	5
36	2	3	2	2	3
110	3	3	2	2	2
122	3	3	3	2	2

Figura 5. Matriz de adjacências correspondente ao grafo virtual.

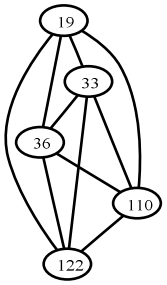


Figura 6 Clique para o grafo virtual na Figura 4.

4.3.2. Cálculo da Clique Máxima no Grafo Virtual.

Este processo visa identificar quais os vértices de decisão são os mais relevantes para o funcionamento do programa que está sendo modelado. Para esta etapa foi aplicada uma metodologia tradicional de identificação da clique máxima [Konc e Janezic 2007] ao grafo virtual gerado na etapa anterior. A clique virtual, gerado a partir da matriz de adjacências da Figura 5, é apresentado na Figura 6.

Como consequência, vértices que corresponderem a elementos desconexos do grafo e vértices que tiverem baixa conectividade com os demais serão desconsiderados.

4.3.4 Redução final do Grafo de Dependência

Na etapa final do processo de redução do grafo de dependência, a lista de vértices pertencentes à clique virtual é usada para determinar se os vértices e arestas presentes no grafo deverão ou não permanecer na versão final do grafo de dependência reduzido.

Para continuar fazendo parte da versão final do grafo de dependência reduzido, o vértice deve possuir um caminho no grafo reduzido original que o ligue até um dos vértices presentes na clique do grafo virtual. Caso ele não possua este caminho, este vértice e todas as arestas associadas devem ser eliminados. Um exemplo do resultado deste processo é ilustrado na Figura 7, que apresenta a versão final do grafo de dependência reduzido do *malware* W32.Evol apresentado na Figura 3. Esta nova versão reduzida será usada para identificação de versões metamórficas deste mesmo *malware*.

4.4. Comparação dos Grafos de Dependência

Os algoritmos de comparação de grafos visam encontrar uma solução através da construção iterativa de associações estabelecidas entre os vértices de dois grafos, G_1 e G_2, que satisfaça um conjunto de restrições do problema em análise. Neste trabalho, o algoritmo de comparação de grafos tem como objetivo gerar uma solução viável para o

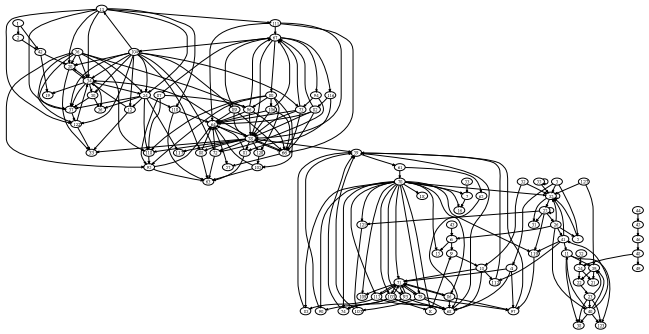


Figura 7. Versão final do grafo de dependência reduzido do W32.Evol.

problema de isomorfismo de grafos. Para os testes da metodologia foram utilizados os resultados obtidos através de três algoritmos distintos. O primeiro é um algoritmo genético tradicional, com tamanho da população constante de 100 elementos, onde 20 novos elementos, gerados com a taxa de *crossover* de 90% e uma taxa de mutação de 20%, substituem os 20 piores cromossomos existentes na geração anterior. Este processo é repetido 2000 vezes antes que o resultado final seja registrado. Os outros são implementações de duas heurísticas propostas por Kim e Moon [Kim e Moon 2010], mas com os seus resultados tratados de forma independente. A fórmula para realizar o cálculo da diferença entre os grafos G_1 e G_2 é definida na Tabela 1.

Os resultados gerados pela execução de cada algoritmo são apresentados através do cálculo da similaridade entre os grafos comparados, gerando uma pontuação de similaridade (ver Tabela 1). A menor dentre as três pontuações geradas, é então utilizada como resultado final da comparação entre G_1 e G_2, passando a ser reconhecida com a pontuação de similaridade entre os grafos comparados. Esta pontuação é usada para determinar se existe ou não contaminação por *malware* no programa analisado.

Tabela 1. Equações para realizar o cálculo da similaridade entre dois grafos

Descrição	Equação
Definições iniciais	$G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ e $\lvert V_1 \rvert < \lvert V_2 \rvert$
Função de busca de uma aresta *e* em um conjunto de arestas *E*	$I(e,E) = \begin{cases} 0, & se \quad e \in E \\ 1, & caso \quad contrário \end{cases}$
Cálculo da similaridade entre G1 e G2	$similaridade(G_1, G_2) = \frac{\sum_{e \in E_1} I(e, E_2) + \sum_{e \in E_2} I(e, E_1)}{\lvert E_1 \rvert}$

5. Avaliação e Resultados Experimentais

Para avaliar a metodologia proposta, foram utilizadas versões metamórficas dos *malware* W32.Evol e W32.Polip. As amostras metamórficas do W32.Evol foram as mesmas utilizadas por Cozzolino et al. [Cozzolino et al. 2012]. No caso do W32.Polip, as amostras foram coletadas em uma base pública [Offensive Computing 2013].

5.1 Coeficiente de Ajuste da Pontuação

Para a definição de um limite máximo da pontuação de similaridade que seria utilizado como o limiar de identificação, foi criada uma base de grafos sintéticos gerados a partir de uma metodologia usada na avaliação de algoritmos para detecção de subgrafos [Conte et al. 2007]. Com base nos resultados obtidos, foi definida uma média de 25% de

similaridade como um nível mínimo para a indicação de contaminação, ou seja, a pontuação do cálculo de diferença observada entre os pares de grafos não deveria ultrapassar o limiar de 0,8 pontos. Entretanto, quando o vírus W32.Evol foi inicialmente avaliado, todos os resultados de pontuação foram superiores a 0,95, o que é muito maior que os 0,8 pontos esperados.

Após analisar tanto os conjunto de grafos sintéticos quanto os grafos gerados com base no W32.Evol, foi identificado que na base sintética a quantidade de vértices de cada par de amostras era sempre a mesma e no caso dos grafos do W32.Evol, a quantidade de vértices variou de 27,45% a 155,56% maior quando comparada com a versão do grafo definido com base de comparação do *malware*, o que explicou a diferença na pontuação obtida inicialmente.

Para tratar este problema, foi necessário aplicar um coeficiente de ajuste da pontuação de similaridade que levasse em consideração a diferença na quantidade de vértices entre os grafos comparados. Após o levantamento dessas diferenças observou-se que, em média, os grafos associados às versões metamórficas possuíam 1,67 vértices a mais que o grafo associado ao código original. Este valor foi então aplicado aos resultados iniciais como um coeficiente redutor, segundo a fórmula $PSA = PSO/r$, onde "*PSA*" é a pontuação de similaridade ajustada, "*PSO*" é a pontuação de similaridade original e "*r*" representa o coeficiente de redução. Esta nova sequência de valores ficou abaixo da pontuação definida como limiar para identificação do *malware* procurado. O resultado deste ajuste é ilustrado na Figura 8. Todos os dados apresentados nos próximos gráficos têm o coeficiente de redução correspondente já aplicado.

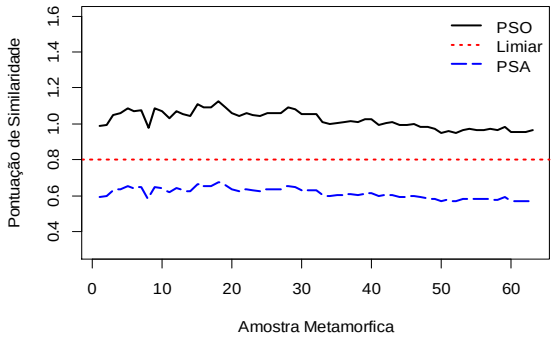


Figura 8. Utilizando o coeficiente de redução na pontuação de similaridade de referência para W32.Evol.

5.2 Resultados com o Processo de Redução Aprimorado

O primeiro conjunto de testes foi baseado no *malware* W32.Evol. Inicialmente, um grafo base para comparação foi gerado a partir de uma versão ainda sem alterações metamórficas do W32.Evol. A seguir, as 63 amostras metamórficas disponíveis foram comparadas diretamente com este grafo base. A pontuação obtida apresentou um coeficiente de variação (desvio padrão dividido pela média) de 4,65%. Testes realizados sobre o mesmo conjunto de elementos, avaliados por outro método [Cozzolino et al. 2012], obtiveram um coeficiente de variação de 47,29%.

Para avaliar a metodologia de redução aprimorada proposta, o arquivo contendo grafo de dependência usado como base de comparação foi então submetido a todo o processo descrito na seção 4.3. Como resultado, a quantidade de vértices do grafo base caiu em 23,52%, passando de 153 para apenas 117 vértices. Finalmente, depois de coletados os resultados da comparação deste novo arquivo reduzido com as 63 versões metamórficas, foram obtidos os resultados apresentados na Figura 9.a, onde são comparados com os resultados da metodologia de referência [Kim e Moon 2010], cujo processo de redução empregado é apenas aquele descrito na seção 4.2. Os resultados obtidos tanto pelo uso da metodologia de referência como pelo uso do processo de redução aprimorado, obtiveram sucesso na identificação de todas as amostras metamórficas, atingindo coeficientes de similaridade abaixo do limiar de identificação definido.

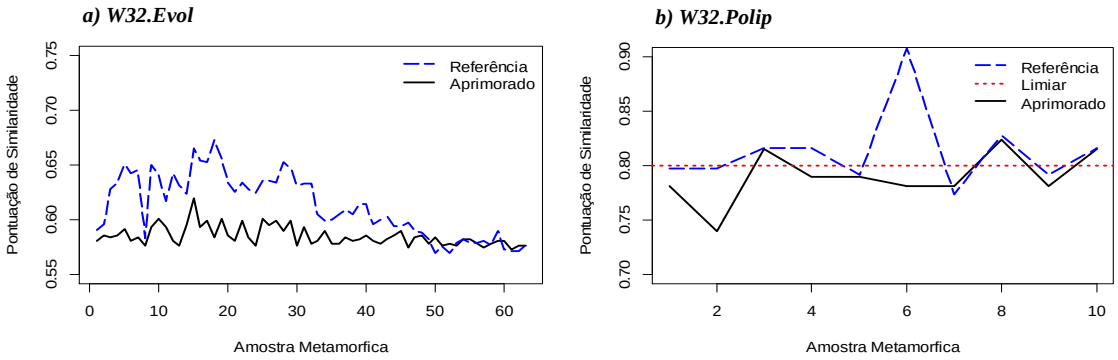


Figura 9. Comparação entre os resultados da metodologia de referência e aqueles obtidos pelo processo de redução aprimorado para os códigos maliciosos W32.Evol (a) e W32.Polip (b).

Entretanto, os resultados obtidos com o uso do processo de redução aprimorado possuem um coeficiente de variação de apenas 1,51%, demonstrando que este processo manteve os componentes do grafo base associado aos elementos mais relevantes do programa a partir do qual foi gerado, diminuindo consideravelmente a interferência que as estruturas eliminadas do grafo de dependência introduziram nos resultados obtidos através da metodologia de referência.

Um segundo conjunto de testes foi realizado com onze amostras do *malware* metamórfico W32.Polip. A quantidade de amostras disponíveis não era muito grande, mas foi possível selecionar uma delas como elemento de referência para a comparação com as demais amostras. Uma dificuldade específica neste conjunto de dados foi o cálculo do coeficiente de redução. A diferença na quantidade de vértices entre o grafo de referência e algumas amostras era muito grande. Uma das amostras, por exemplo, possuía mais de quatro mil vértices, enquanto o grafo de referência possuía apenas 23 vértices. Se estas amostras entrassem no cálculo, o valor gerado seria um coeficiente de redução superior a 20, o que distorceria os resultados. Assim, duas amostras que possuíam mais de dois mil vértices foram desconsideradas, deixando como coeficiente de redução final o valor de 1,19 para os resultados de comparação do W32.Polip.

Os resultados finais da avaliação do W32.Polip são apresentados na Figura 9.b. Neste conjunto de dados a metodologia de referência identificou apenas 50% das amostras metamórficas, enquanto que o uso do processo de redução aprimorado

permitiu a identificação de 70% destas mesmas amostras. Novamente tivemos uma melhoria na estabilidade dos resultados produzidos, apesar de não se ter observado uma diferença tão significativa quanto no primeiro conjunto de testes, já que a o coeficiente de variação correspondente à metodologia de referência foi de 4,51%, contra um coeficiente de variação de 3,05% para a metodologia de redução aprimorada.

6. Conclusões

Neste trabalho foi proposta uma abordagem destinada a identificar *malware* metamórfico através da comparação de grafos de dependência armazenados em uma base de referência, onde foi aplicado um processo de redução aprimorado baseado na diferenciação de nós em conjunto com a utilização de estruturas virtuais. Embora os dados experimentais com a base sintética de teste apresentaram características que não puderam ser observadas nos dados iniciais obtidos da avaliação de programas reais, a utilização de um coeficiente de redução, diretamente relacionado com a diferença entre o número de vértices dos grafos avaliados, tornou a pontuação consistente com o modelo de comportamento esperado.

A utilização de estruturas temporárias, como as arestas virtuais e a geração da clique virtual, forneceu uma maneira gerenciável de lidar com as características dos grafos de dependência, mapeando a relação estrutural entre os componentes mais relevantes dos grafos. Isto permitiu que o processo de redução aprimorado diminuísse o tamanho dos grafos de dependência da base de referência, sem prejuízo para sua utilização no processo de identificação. Nos testes com os *malware* W32.Evol e o W32.Polip, houve diminuição na variância dos resultados e, no caso do W32.Polip também foi observada uma melhoria nos resultados de identificação. Assim, os resultados obtidos em todos os testes mostram o potencial da abordagem proposta na melhoria do processo de identificação de código metamórfico.

As próximas etapas serão dedicadas à modificação do cálculo de pontuação de similaridade, eliminando a necessidade do coeficiente de redução, e o desenvolvimento de uma nova abordagem para resolver o problema do isomorfismo máximo de subgrafo entre grafos de dependência, com foco na redução do tempo de processamento necessário. Esta nova abordagem deve aproveitar ao máximo as informações sobre o tipo de vértice, em conjunto com o uso de ordenação topológica, já que o primeiro permite um processo de comparação seletivo, que manipule apenas elementos de mesma natureza e o último permite organizar a estrutura do grafo de uma forma que está mais relacionado com a ordem de execução das instruções do programa original.

Agradecimentos

Este trabalho foi financiado pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) através do processo 062.03178/2012 (Edital Universal Amazonas). Agradecemos ainda a FAPEAM e a CAPES pelo apoio financeiro com bolsa de doutorado.

Referências

Baker, W. Hutton, A. Hylender, C. D. Pamula, J. Porter, C. e Spitler, M. (2011). 2011 data breach investigations report. Verizon RISK Team. http://www. verizonbusiness.com/resources/ reports/rp_databreach-investigations-report-2011_en_xg.pdf.

Bomze, I. M. Budinich, M. Paradalos, P. M. e Pelillo, M. (1999). “The maximum clique problem”. Handbook of combinatorial optimization. Springer US, p. 1-74.

Borello, J. e Mé, L. (2008). “Code obfuscation techniques for metamorphic viruses”, Journal in Computer Virology, vol. 4, núm. 3, pág. 211-220.

Bruschi, D. Martignoni, L. e Monga, M. (2007). "Code Normalization for Self-Mutating Malware" , IEEE Security & Privacy, v. 5, n. 2, p. 46-54.

Conte, D. Foggia, P. e Vento, M. (2007). “Challenging Complexity of Maximum Common Subgraph Detection Algorithms: A Performance Analysis of Three Algorithms on a Wide Database of Graphs”,J. Graph Algorithms Appl, v. 11,n. 1, p. 99-143.

Cozzolino, M. Martins, G. Souto, E. e Deus, F. (2012). “Detecção de Variações de Malware Polimórfico por Meio de Normalização de Código e Identificação de Subfluxos”, Anais do XII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Curitiba. Sociedade Brasileira de Computação, p. 30-43.

Ferrante, J. Ottenstein, K. J. e Warren, J. D. (1987). “The program dependence graph and its use in optimization”, ACM Transactions on Programming Languages and Systems (TOPLAS), v. 9, n. 3, p. 319-349.

Floyd, R. (1962). “Algorithm 97: shortest path”. Communications of the ACM 5.6, 5:345.

Garey, M. R. e Johnson, D. (1979). “Computers and Intractability: A Guide to the Theory of NP-Completeness”, W. H. Freeman & Co.

Griffin, K. Schneider, S. Hu, X. e Chiueh, T. C. (2009). “Automatic Generation of String Signatures for Malware Detection”. Proceedings of the 12th Symposium on Recent Advances in Intrusion Detection (RAID2009), Brittany, França, p. 101-120.

Hsiao, S.-W. Sun, Y. S. Chen, M. C. e Zhang, H. (2010). "Behavior Profiling for Robust Anomaly Detection," IEEE International Conference on Wireless Communications, Networking and Information Security, p. 465-471.

Hu, X. Chiueh, T.-c. e Shin, KG. (2009). “Large-scale malware indexing using function-call graphs”. Proceedings of the 16th ACM conference on Computer and communications security, p. 611-620.

Jacob, G. Debar, H. e Filiol, E. (2009). “Malware Detection using Attribute-Automata to parse Abstract Behavioral Descriptions”, CoRR, abs/0902.0322.

Karin, A. (2006). “Automatic Malware Signature Generation”. 16 de Outubro, 2006. Disponível em http://web.it.kth.se/~cschulte/teaching/theses/ICT-ECS-2006-122.pdf.

Kim, K. e Moon, B. (2010). “Malware Detection based on Dependency Graph using Hybrid Genetic Algorithm”. Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, p. 1211-1218.

Konc, J. e Janezic, D. (2007). “An improved branch and bound algorithm for the maximum clique problem”. Communications in Mathematical and in Computer Chemistry, n. 58, p. 569-590.

Liu, C. Chen, C. Han, J. e Yu, P. S. (2006). “GPLAG: detection of software plagiarism by program dependence graph analysis”. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 872-881.

Moura, A. V. e Rebiha, R. (2009). “Automated Malware Invariant Generation”, International Conference on Forensic Computer Science (ICoFCS), p. 7.

Notoatmodjo, G. (2010). “Detection of Self-Mutating Computer Viruses”, Disponível em http://www.cs.auckland.ac.nz/compsci725s2c/archive/termpapers/gnotoadmojo.pdf, Department of Computer Science, University of Auckland, New Zealand.

Offensive Computing. (2013). “Offensive Computing”. http://www.offensivecomputing.net/. Nov. 2013.

Warshall, S. (1962). “A theorem on boolean matrices”. Journal of the ACM (JACM), n. 9, p. 11-12.

An Ontological Approach to Mitigate Risk in Web Applications

Marcius M. Marques, Célia G. Ralha

Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Brasília – DF – Brazil

marcius@marciusmarques.com, ghedini@cic.unb.br

Abstract. *Information Security (InfoSec) is becoming a high priority asset to support business activities, as organizations struggle to assure that data is available and secure in web applications. However, security is not a concern from the beginning of the development process, mainly because developers are not security specialists. Consequently, vulnerable systems are designed and when attacked can compromise organization's data and operations, enclosing high financial losses. Because most attacks targets the application layer, we propose an intelligent approach based on ontology to mitigate risks in web applications. An ontological approach can contribute to InfoSec knowledge dissemination and reduce the burden of implementing secure web applications on organizations. The ontology is based on the OWASP Top 10 Project, applied to reduce the gap between the application developer and the security knowledge. The proposed model is employed in the development's design phase; with more secure web applications as the outcome. The extensible and reusable developed ontology is evaluated in a prototype scenario of a web application named 'SMS Broadcast'. The results show that vulnerabilities can be reduced by increasing the security awareness of web developers during the application development process.*

1. Introduction

The fact that organizations dependability on information systems (IS) to manage their business activities is increasing is a known and irreversible one [Weske 2007]. Global networking and Information Technology (IT) advances at high speeds allowing all types of organizations to take advantage and achieve better results, in order not to risk falling behind competitors. IT is a strategic area for organizations, with web-based systems playing a central role in the modern economy, where information needs to have instant availability. This needed feature is followed by an increase in the number and sophistication of attacks to web applications, and as a result, organizations faces a great challenge to keep information secure [da Silva and Ellwanger 2012].

Although Information Security (InfoSec) is a growing spending priority in most organizations, the vulnerability rates and losses numbers are very high. In a security assessment with more than 200 web applications (including e-commerce, on-line banking, credit cards companies, etc), vulnerabilities that could be explored were found in more than 90% of them [Only 10% 2005]. In Cyberattacks (2013), it is reported that companies losses due to hacking and cybercrime range from U$300 billion to U$1 trillion dollars, with hackers stealing more than one terabyte of data daily from vulnerable web applications. According to Key Findings (2013), the attacks on web

applications are a routine part of business and will be a part of doing business going forward. The worrying potential economic impact related to InfoSec for organizations can be found in details at Gordon and Loeb (2002).

A successful InfoSec program enfolds many layers, including software to detect viruses, firewalls, sophisticated encryption techniques, intrusion detection systems, automated data backup and hardware devices, to name a few. Organizations have then to decide what exactly needs to be protected, what is the level of protection that each resource requires and which tools can be used to achieve it all [Almeida 2007]. Achieving consensus regarding safeguards for an IS, among different stakeholders in an organization, has become more difficult than solving many technical problems [Dhillon and Backhouse 2000]. Moreover, the needed knowledge to apply a successful InfoSec project is available on technical standards (e.g. ISO 27001) or in the head of security specialists. Consequently, InfoSec projects tend to be complex, expensive and time consuming, with unclear and hard to measure benefits [da Silva et al. 2011].

However, researches showed that 75% of attacks are being deployed at the application layer vice infrastructure [Razzaq et al. 2009], giving opportunities for risk mitigation within the control of organizations. Because software developers are usually not security specialists, web applications are designed with minor or none security concerns. The neglect of good programming practices, including the simplest ones, is one of the main causes for the existence of vulnerabilities in web-based systems [Silva and Ellwanger 2012].

In this article, we propose an intelligent approach that uses an ontology to reduce the gap between web application developers and the needed security knowledge. The source of information is based on the OWASP (Open Web Application Security Project) initiative, specifically on the OWASP TOP10 Project [OWASP Top 10 Project 2014]. This project is constantly updated with the most critical web applications security flaws. When adopted by an organization, the main target is to change the software development culture in order to produce secure code.

One of the core OWASP project pillars that we agree with is that security in application development should be considered since the beginning of the development process, being included in all stages. Nevertheless, in this work we intend to focus on the design phase of the Software Development Life Cycle (SDLC). By increasing the security awareness of the web application developer, we believe the final product will be more secure as known potential risks will be mitigated.

The proposal is related to the risk management aspect of security activities, a very important process within InfoSec, required to be part of the organizations' security policy [Peltier 2013]. From the definition in Whitman and Mattord (2011), Risk Management (RM) is the process of identifying risk, represented by vulnerabilities, to an organization's information assets and infrastructure, and taking steps to reduce them to an acceptable level. RM involves three well-defined steps: identification, assessment and control, which will be the focus of the ontology we built to achieve our goal. By using this ontology, a web developer who is not a security specialist is able to identify and mitigate the risks related to the web application during the design phase, being this approach the main article's contribution.

The rest of the document is organized as follows. In Section 2 is presented the relation between ontology and InfoSec; in Section 3, the OWASP Top 10 Project used

to define the ontology is discussed; in Section 4 the solution proposal is showed, with the evaluation case results on Section 5. In Section 6, related work is listed and finally in Section 7 the conclusions and future work are presented.

2. Ontology and Information Security

Ontologies are being extensively used in different fields of study, primarily to organize information and formalize knowledge. It is receiving a special and growing attention from Computer Science professionals as experiences in IS development have shown to be related to long and expensive processes [Bai and Zhou 2011].

In Grubber (1993), there is an initial definition that is largely acceptable when ontology is related to Computer Science – "ontology is an explicit specification of a contextualization". This definition has been evolving over time. Another numerous times referenced definition is the one in Guarino (1998) – "an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary". Other definitions can be found, most of them complementing each other's meaning as the two mentioned before. For the purpose of this article, ontology will be a tool for InfoSec knowledge representation and organization.

An overview can be found at Almeida and Bax (2003), where ontologies are classified considering many different aspects like function, applicability, structure and contents. Depending on the level of abstraction, it can be divided in four groups: high-level, domain, task and application ontologies. High-level (or foundational) ontologies are based on very high generalization concepts, so it can be applied in different domains. Domain ontologies describe the vocabulary related to a defined domain, by specializing the concepts from foundational ontologies. In this article, we built a domain ontology, using InfoSec concepts as the domain, more specifically the OWASP Top 10 Project knowledge to secure web application development.

According to Raskin et al. (2001), the use of ontology in InfoSec can be summarized in one of two possible approaches: Based on Natural Language Processing - a reactive approach where an ontology is used to aid in the management of the huge volume of data provided by security logs and vulnerabilities alerts. The ontology is constantly updated with new information that is then used in attack analysis and prevention. In this method, the ontology is usually combined with other tools to provide a unified solution; and Based on Knowledge Representation – a proactive approach where an ontology is built to gather security domain concepts in order to help stakeholders to make security related decisions, according to organization's requirements. The ontology can be defined in different levels of abstraction and used for different objectives related to security activities.

Our proposal belongs to the second category – a proactive approach - information for risk mitigation in web applications will be modeled using domain ontology so application developers can use it during the design phase. The source of information for the ontology is a free open-source project that is detailed in Section 3 – the OWASP Top 10.

Considering the benefits of adopting the InfoSec domain ontology based on knowledge representation at the organization scope, we cite: i) creation of conceptual models to better understand security incidents; ii) support the interoperability between

different security tools; iii) creation of a standard for structuring security data, allowing terms to be mapped to the ontology; iv) use of automatic queries and inferences to filter ontology information. Moreover, the approach can also benefit from the basic ontologies capabilities of reuse and scalability [Almeida et al. 2010].

3. OWASP Top 10 Project

OWASP was established as an international organizational in 2004. It works as an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted [About OWASP 2014]. Because it is free from commercial and governmental relations, it is able to provide unbiased, practical, cost-effective information about application security, producing many types of materials in a collaborative and open way.

The InfoSec materials in OWASP are usually organized into independent open projects and currently there are almost 200 active projects. The TOP 10 Project is the most popular project at OWASP initiative, with the first version released in 2003. Subsequent releases happened in 2004 and 2007 with minor adjustments.

The 2010 OWASP TOP 10 version was the first to be prioritized by risks, as it is in the latest 2013 version. It lists the TOP10 security issues based on data from seven application security companies, which includes information from thousands of organizations and applications. The risks are rated using the rating scheme presented in Table 1, which is the basis for our ontology classes' definition.

Table 1. Rating scheme for OWASP TOP10 (from OWASP TOP 10 Project, 2014)

Threat Agents	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
App Specific	Easy	Widespread	Easy	Severe	App / Business Specific
	Average	Common	Average	Moderate	
	Difficult	Uncommon	Difficult	Minor	

The current security risks in the latest 2013 version are: A1 - Injection, A2 – Broken Authentication and Session Management, A3 – Cross-site Script (XSS), A4 – Insecure Direct Object Reference, A5 – Security Misconfiguration, A6 – Sensitive Data Exposure, A7 – Missing Function Level Access Control, A8 – Cross-site Request Forgery (CSRF), A9-Using Component with Known Vulnerability and A10 – Unvalidated Redirects and Forwards. Each risk is classified according to the rating scheme presented in Table 1, and the final document includes the following information: how to find out if applications are vulnerable to the risk, how to prevent the risk, risk examples and information references.

There are different initiatives to classify and discover vulnerabilities, most of them being supported by companies in the best interest to solve security issues. Organizations like SANS Institute (www.sans.org) and Mitre Corporation (www.mitre.org) provides the CWE (Common Weadness Enumeration) about the TOP 25 most dangerous software errors, listing the problematic practices in different categories [CWE/SANS TOP 25 2011]. We choose OWASP Top10 among other sources as it has a more didactic structure that can help developers and eases the ontology construction process. OWASP has more often updates and it is accepted by

many patterns as the minimal security requirements for web applications [PCI 2009]. The OWASP Top10 is also connected with other OWASP projects so they can be integrated in the future by using the ontology scalability property (example: OWASP Testing Guide and OWASP Risk Rating Methodology).

4. Proposal

This work proposal includes the OWASP TOP10 ontology that can be used by someone who is not a security specialist to evaluate what are the risks related to the web application being developed. For the ontology development, we use the method known as *101 Methodology* [Noy and McGuiness 2001]. The tool employed to build the ontology is the Protégé, a free open-source ontology builder from Stanford University (http://protege.stanford.edu/), and the language is the OWL (Web Ontology Language). The ontology uses definitions from the OWASP Top 10 scheme only, in order to have conceptual consistence, but it can be extended in the future to include new security aspects, like the ones related to network topology, for instance.

To illustrate the OWASP Top10 ontology, let us take the Risk A1 - Injection, where internal and external users are considered threat agents. Considering the rate scheme presented in Table 1, the attack vector is the own application interpreter that can receive malicious text-based attacks, classified to be easy to explore. The security weakness exists in SQL, LDAP, XPath and other technologies, prevalence is common and the technical impact for the organization is classified as severe.

For each of the terms presented in Section 3 for the OWASP TOP10 Project scheme, we created a class in the ontology, related to the super class 'Risk'. For each class, the data property and type presented in Table 2 were defined. Each data property is related to the scheme design presented in Table 1.

Table 2. OWASP TOP10 Ontology classes and data properties

Class	Data Properties	Type
Risk	riskName	String
	riskRange	String
ThreatAgent	threatAgentDescr	String
AttackVector	Exploitability	Easy / Average / Difficult
	attackVectorDescr	String
SecurityWeakness	Prevalence	Widespread / Common / Uncommon
	Detectability	Easy / Average / Difficult
	securityWeaknessDescr	String
Impact	technicalSeverity	Severe / Moderate / Minor
	technicalDescr	String
	businessDescr	String
Control	controlDescr	String

The relationships between classes were created using the object property feature of Protégé, based on InfoSec concepts for each of the classes retrieved from Whitman and Mattord (2011). A graphic view created with the *OntoGraf* built-in feature from the Protégé tool is shown in Figure 1.

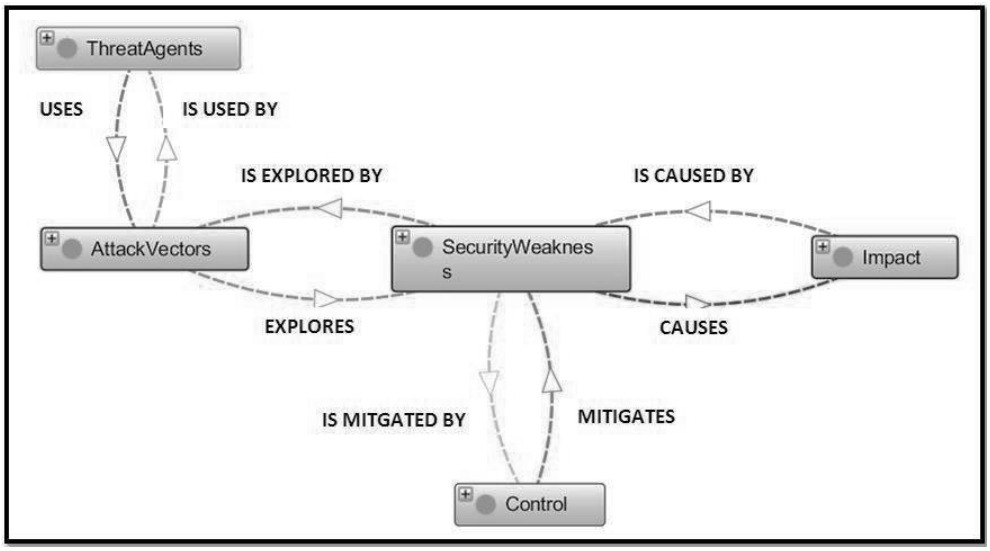


Figure 1. OWASP TOP10 Ontology classes diagram and relationships

Each individual of the ten-listed Risk (as presented in Section 3) is associated with an individual of each other class, according to the different object properties defined in Figure 1. For example, when Risk1 is instantiated as an individual, it is related to ThreatAgent1, AttackVector1, Control1, Impact1 and SecurityWeakness1. Following the model presented in Figure 1, ThreatAgent1 *uses* AttackVector1. AttackVector1 *explores* SecurityWeakness1. SecurityWeakness1 *causes* Impact1. Finally, Control1 *mitigates* SecurityWeakness1. An example of the individual AttackVector1 related to Risk A1-Injection is illustrated in Figure 2.

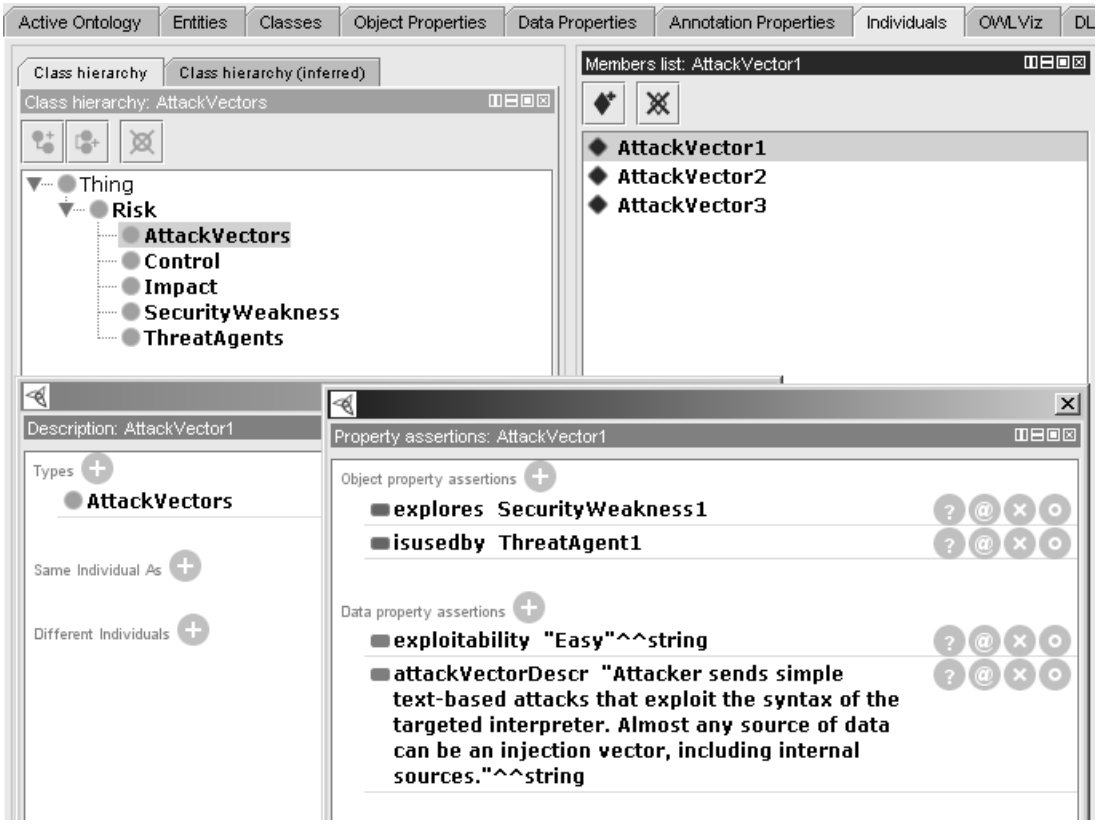


Figure 2. OWASP TOP10 Ontology AttackVector1 individual

The OWASP TOP10 ontology consistency was verified by using the Protégé Pellet 2 plugin [Clarck&Parsia 2011] as a reasoning engine. It checks for hierarchies, domains, ranges, conflicting disjoint assertions, and others issues through all ontology.

Once the ontology is consistent, information can be recovered through the use of a query language based on RDF (Resource Desciprtion Framework) such as SPARQL (SPARQL Protocol and RDF Query Language) [Pérez et al. 2006]. RDF is a query language designed to be applied on a set of "subject-predicate-object" triples, very similar to SQL (Structured Query Language). SPARQL is widely used as a powerful and complementary tool to ontologies, especially when used in the Semantic Web for knowledge representation.

The TOP 10 ontology is aligned with the basic preliminary criteria that must be taken into consideration before building an ontology - clarity, consistency and extensibility, as stated in Grubber (1993). Consider that the individuals defined in the ontology are the ten risks listed in Section 3, with data properties completed according to the OWASP TOP10 Project available information. It encompasses all the concepts available from the data used as the source of information in order to be applied in risk management activities of secure web application development as presented in Section 5.

5. The development of secure web application

The OWASP TOP10 Ontology was tested in a real case scenario during the development of a web application named ‘SMS Broadcast’. This application is designed to send text messages to registered organization’s employees based on selection filters, in an eventual emergency situation that requires employees to be notified immediately and at once. The text message is sent to a third-party bulk SMS provider that delivers it for the users according to the system’s configuration. The chosen application includes most of the commonly found classes of web-based systems like authentication, parameters communication and privilege level to name a few.

To comply with organization requirements, the chosen language for the application is classic ASP (Active Server Pages) and the publishing web tool is Microsoft SharePoint®. The web application was designed to have two modules:

- Module 1 – Users Registration – all organization’s users must be able to apply to receive text message in case of emergencies. When applying, they need to provide information for the following self-explanatory fields: *lastname*, *firstname*, *city*, *section* and *phonenumber*
- Module 2 – Message Broadcast – selected organization’s employees will have access to this page where they can see all registered users and select the ones they want to receive the message. For example, it should be possible to select all employees from *financial* section in a specific city and send a message to them only.

The evaluation of the proposed model uses two different scenarios.

Scenario 1 - the task to develop the web application was assigned to two web developers – DA1 and DB1. Both have similar work experience and knowledge, except that web developer DA1 has attended security courses that included OWASP awareness activities in the last year. As a result, two different web applications were created, one coded by developer DA1 and one coded by developer DB1.

Scenario 2 - two others developers, DA2 and DB2, are assigned the same task. Exactly as in Scenario 1, both have similar work experience and knowledge, except that web developer DA2 has attended the same security courses as web developer DA1. The main difference is that in this scenario, during the design process, web developers DA2 and DB2 are required to answer a questionnaire with twenty questions about the application requirements they will develop, and only after receiving the questionnaire output with potential risks they should be worried about, they should start the development process.

The questionnaire has only application specific questions that developers must know how to answer at the design phase. For example: (i) System contains information in transit coded in XML language, (ii) User can recover login information by using 'forgot password/forgot login' feature, (iii) There is sensitive data (PII – Personally Identifiable Information) stored in the system's database. For each question the answer can be "Yes", "No" or "N/A". The questionnaire was prepared by three application developers that are also security specialists with extensive hands on experience on InfoSec. They have been delivering OWASP and InfoSec training for others developers with different levels of experience and knowledge around the world in the last three years within the organization. Due to space limitation, it was not possible to include the complete questionnaire in the article, but it is available in *marciusmarques.com/owasp*.

Based on DA2 and DB2 answers to the questionnaire, an interface that was developed using Apache Jena (https://jena.apache.org/) informs the risks associated with the web application to the developer. The interface will use ARQ, a SPARQL processor for Jena, to query the Top 10 Ontology using SPARQL language. The objective is to advise the web developer about how to mitigate the risks in the design phase, in order to build a more secure code from the beginning of the development process. The proposal architecture is presented in Figure 3.

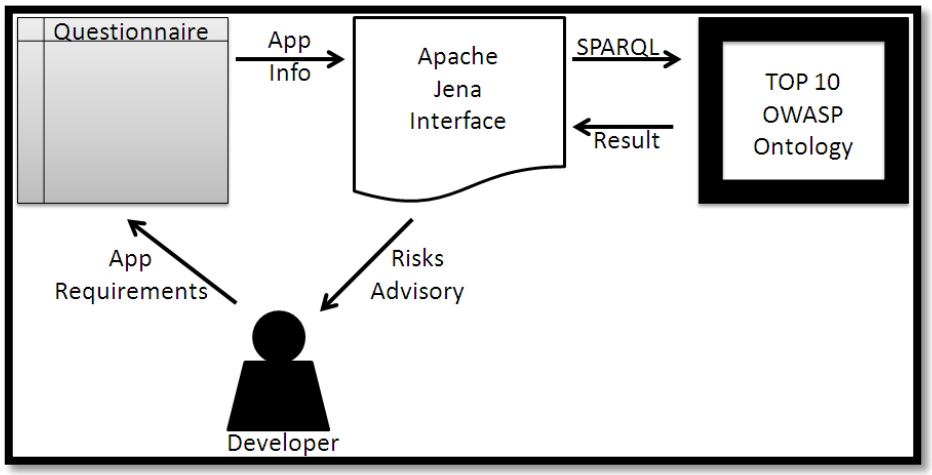


Figure 3. Proposal architecture

To illustrate the proposed model, assume the question "User can click a 'remember me' box so they don't have to re-authenticate" receives a "Yes" answer. The SPARQL query will associate the key-terms (remember me, session, authentication) with the risk A2 (Broken Authentication and Session Management) in the OWASP TOP 10 ontology class, as this information is available mainly at the data property 'securityWeaknessdescr' from risk A2, as presented in Table 2. As a result, control

measures related to this risk A2 available at the data property 'controlDescr' from the OWASP Top 10 ontology will be presented for the developer in the end of the questionnaire which can be used during the development process. Once concluded, web application security of each scenario are evaluated as detailed in Section 5.1.

5.1. Results analysis and discussion

To evaluate the web applications created, the four ASP codes developed (DA1, DB1, DA2 and DB2) for both modules were submitted to the organization's CnA (Certification and Accreditation) process. During the CnA, security specialists analyzes the system using many different criteria related to security aspects. For the purpose of this article, we will present the code review and penetration test score, which are part of the CnA process. Due to the organization's security policy, the score methodology and tools used cannot be detailed in the article. Based on the result of the CnA, the web application can be authorized to go to production when it attends the minimum security requirements. If it does not attend, it needs to be reviewed in a new submission after the changes suggested in the CnA report are performed.

The combination of the code review with penetration test is considered one of the most effective methods to be used during the assessment of the security of an application [Curphey and Arawo 2006]. A comparison and benefits of both can be found in an OWASP conference presentation - The Strengths of Combining (2009).

The metric used by the organization requires that the system achieve a maximum score of six in the CnA security assessment in order to have its implementation authorized. The higher the security risks, the higher the score. The results for the codes submitted and a summary of the scenarios can be found in Table 3.

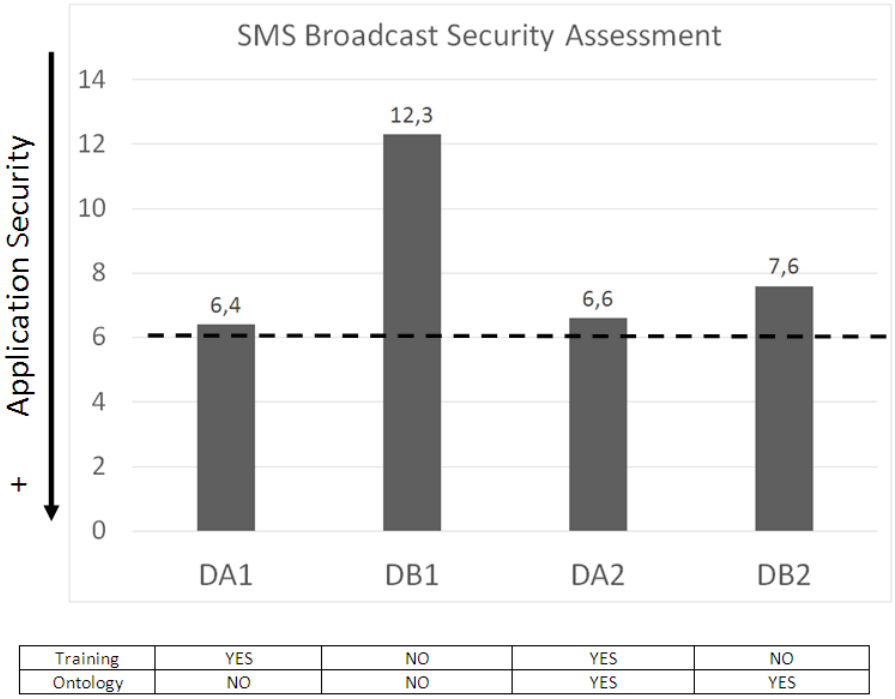
Table 3. Proposed model evaluation scenarios and results

Web Developer	DA1	DB1	DA2	DB2
Development Experience	High	High	High	High
Security awareness	High	Low	High	Low
TOP 10 Ontology use	No	No	Yes	Yes
Development Outcome	DA1 code	DB1 code	DA2 code	DB2 code
Risk assessment CnA score	6,4	12,3	6,6	7,6

From the results, the most vulnerable application was the one developed by the developer DB1 that did not have specific security training about web application development and did not use the proposed OWASP TOP 10 ontology to evaluate the risks associated with the application requirements. Web applications from developers DA1 and DA2, although need to be reviewed as neither reached the minimum score of six, are the most secure ones. Both were developed by someone with a security awareness background, having DA2 using also the OWASP Top 10 ontology during the design phase. Finally, DB2 developer had no security awareness training and by means of the proposed ontology achieved an acceptable score of 7,6 in his web application.

Web developer DA2 and DB2 reported that it was useful to use the questionnaire output before the development coding step had started, as it was helpful to identify in advance security issues they were not considering in the beginning. Both agreed that the information provided by the ontology influenced in the development process in a positive and efficient way. Positive because risks were avoided and

efficient because it was not required going through long reading and analysis activities to identify application related issues. A graphic view of the result is presented in Figure 4.



Training	YES	NO	YES	NO
Ontology	NO	NO	YES	YES

Figure 4. Results overview

6. Related Work

The use of domain ontologies to aid InfoSec related activities can be found with distinct objectives under both approaches: reactive and proactive (Section 2).We analyzed some related work where InfoSec domain ontologies are created with different levels of details using different source of information, aiming to build a knowledge repository that can be useful for the organization. The main challenge of all proposals is how to use the security information efficiently in the organizations.

For instance, in Almeida et al. (2010) and de Azevedo et al. (2007), the ontologies were created mainly to classify the information, in order to point out what needs to be protected and how. This approach is in line with business management activities, where SPARQL can be used to identify more high-level security concerns.

SPARQL querying domain InfoSec ontologies was also used in da Silva et al. (2011) and Martimiano and Moreira (2005). On the later, security incident data was used as the source of information to validate the ontology, which can be combined with other approaches to produce interesting results. This type of approach was proposed in Silva and Ellwanger (2012), where *CODI* ontology is presented. It groups an InfoSec ontology with influence diagrams to create a methodology that could facilitate the dissemination of information and the accumulation of knowledge among stakeholders, but the implementation is suggested as a future work.

A reactive approach that uses ontology can be found in Razzaq et al. (2009). It presents an intrusion detection system developed using an InfoSec ontology, aiming to detect zero-day attacks in the application layer, by comparing the event information with a knowledge base that is updated constantly. The idea presented in Rosa et al. (2011) is similar, where an ontology is suggested to detect XML Injection attacks via web services. In both works, the ontology is applied for scanning and assessing the vulnerabilities when it already exists in the web applications.

The contribution of our work is different from previous mentioned in regards to three main concepts. First, our target audience is well defined, it is focused on web application developers; second, the timing of applying the ontology is the design phase of the software development cycle; third, it uses OWASP Top10 Project as the source to build the knowledge about InfoSec. It is similar to others in the sense that it is based in the knowledge representation approach discussed in Section 2 and it uses SPARQL to query the ontology. However, it does not require any pre-knowledge about SPARQL from the user in order to achieve the results.

7. Conclusion and Future Work

In order to be more efficient, effective and responsive to remain competitive, organizations need to be up to date with networks and computer based IS. Due to the features provided by web applications, it is becoming the favorite choice for organizations in order to offer customers a modern solution to perform business. However, this solution charges a price related to needed security concerns, which are often not in the top priority list of organization's stakeholders. This happens not as a deliberated choice, but because InfoSec is a complex and expensive business process.

The main objective of InfoSec in web applications is to reach a balance between accurate access and secure information. IS must hold data that has to be available for authorized users. On one side, it is possible to grant access to everything to everyone. On the other side, it is possible to remove all computers from the network to prevent unauthorized accesses, losing of course all benefits from information sharing. InfoSec is somewhere between these two utopic realities, with business processes that exists to attend users and system's needs without giving up to all defined security requirements at the same time.

Application developers play an important part in the line of defense of web-based systems, as the majority of current explored vulnerabilities are a consequence of development activities executed without the necessary security concerns. "The security issues are not being adequately considered during the development process, both by lack of knowledge as by the pressure caused by tight delivery schedules" [Uto and de Melo 2009].

Based on this scenario, this article presented an ontological approach to produce more secure web applications. It relies on the application developer common knowledge about the system he is about to develop, in order to provide information on the security risks related to this application. Our goal is to identify the good security practices necessary to be applied in order to mitigate the risks related to the application that is being developed. The burden of training and research is abstracted from the web developers as an ontology is used to provide information about security concerns.

The proposal is tested in a real case scenario, where four developers are assigned the task to build a simple web application with commonly used features for this type of system. They have different knowledge about security aspects and two of them are using the proposed OWASP Top 10 Ontology. After being submitted to the organization's risk analysis CnA process, it was found that the use of the ontology was useful to produce more secure web application. Similar risk scores were achieved by a web developer (DA1) with many hours of security awareness courses when compared to a web developer (DB2) with no security training but making use of the ontology for the same system.

However, none of the four applications could be implemented in the first version, as the minimum score required by the organization was not obtained. This emphasizes the importance of risk assessment activities in the process of web application development, which is in line with the presented proposal of executing it during the design phase. Moreover, both developers that used the ontology reported the benefits of it to mitigate risks in web applications.

The proposed ontological approach does not require that the application developer becomes a security specialist in order to produce a more secure system. It can also benefit from the advantages offered by knowledge representation using ontologies.

As future work, we suggest the use of different inference tools compatible with the OWL standard to query the ontology; the integration of the TOP 10 ontology with other InfoSec domain ontologies to produce other results and the use of other sources of information for ontology instantiation according to the organization's requirements (e.g. ISO27001). In the meantime, we are executing a similar test with another more complex web application to compare the results with the ones from this article. We also intend to expand the ontology by using the information provided in other OWASP Project – SAMM (Software Assurance Maturity Model) [OWASP SAMM Project 2013]. We believe a more comprehensive model can have even more significant results.

8. References

About OWASP. (2014). Retrieved June 10, 2014, from https://www.owasp.org/index.php/About_OWASP

Almeida, M. B. (2007). Aplicação de ontologias em segurança da informação. Diretoria da Prodemge. Revista Fonte, 4(7),75-83.

Almeida, M. B., and Bax, M. P. (2003). Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. Ciência da Informação, Brasília, 32(3), 7-20.

Almeida, M. B., Souza, R. R., and Coelho, K. C. (2010). Uma proposta de ontologia de domínio para segurança da informação em organizações: descrição do estágio terminológico. Informação & Sociedade: Estudos, 20(1).

Bai, X., and Zhou, X. (2011). Development of Ontology-Based Information System Using Formal Concept Analysis and Association Rules. In Advances in Computer Science, Intelligent System and Environment (pp. 121-126). Springer Berlin Heidelberg.

Clark and Parsia (2011). “Pellet: OWL 2 Reasoner for Java”. Retrieved June 18, 2014, from http://clarkparsia.com/pellet/protege/

Curphey, M., and Arawo, R. (2006). Web application security assessment tools. Security & Privacy, IEEE, 4(4), 32-41.

Cyberattacks. (2013). Retrieved June 10, 2014, from http://www.cnet.com/news/cyberattacks-account-for-up-to-1-trillion-in-global-losses/

CWE/SANS TOP. 25 (2011). Retrieved June 11, 2014, from http://www.sans.org/top25-software-errors/

da Silva, B. A., Ellwanger, C. (2012). CODI Methodology for Managing Security in Web Application Development.

da Silva, P. F., Otte, H., Todesco, J. L., and AO, F. (2011). Uma ontologia para gestão de segurança da informação. In: IV Seminário de Pesquisa em Ontologia no Brasil (p. 141).

de Azevedo, R. R., Almeida, M. J. S., and Barros Filho, C. (2007). Uma Ontologia Genérica de Segurança Aplicada a Gestão de Processos de Negócios. In: I Workshop Brasilieiro em Gerenciamento de Processos de Negócios (WBPM).

Dhillon, G., and Backhouse, J. (2000). Information System Security Management in the New Millennium. Communications of the ACM, 43(7), 125.

Guarino, N. (Ed.). (1998). Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy (Vol. 46). IOS press.

Gordon, L. A., and Loeb, M. P. (2002). The economics of information security investment. ACM Transactions on Information and System Security (TISSEC), 5(4), 438-457.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220.

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., and Booch, G. (1999). The unified software development process (Vol. 1). Reading: Addison-Wesley.

Key Findings. (2013). Retrieved June 11, 2014, from http://www.pwc.com/en_US/us/increasing-it-effectiveness/publications/assets/us-state-of-cybercrime.pdf

Martimiano, L. A., and Moreira, E. S. (2005). Using ontologies to assist security management. In Proceedings of the 8th International Protégé Conference.

Noy, N. F., and McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology, from http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

Only 10%. (2005). Retrieved June 10, 2014, from http://www.prnewswire.com/news-releases/only-10-of-web-applications-are-secured-against-common-hacking-techniques-58703902.html

OWASP TOP 10 Project. (2014). Retrieved June 10, 2014, from https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP SAMM Project. (2013). Retrieved June 10, 2014, from https://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model

PCI (2009). Payment Card Industry (PCI) Data Security Standard. Requirements and Security Assessment Procedures, version 1.2.1. PCI Security Standards Council.

Peltier, T. R. (2013). Information Security Policies, Procedures, and Standards: guidelines for effective information security management. CRC Press.

Pérez, J., Arenas, M., and Gutierrez, C. (2006). Semantics and Complexity of SPARQL. In The Semantic Web-ISWC 2006 (pp. 30-43). Springer Berlin Heidelberg.

Raskin, V., Hempelmann, C. F., Triezenberg, K. E., and Nirenburg, S. (2001). Ontology in information security: a useful theoretical foundation and methodological tool. In Proceedings of the 2001 workshop on New security paradigms (pp. 53-59). ACM.

Razzaq, A., Ahmed, H. F., Hur, A., and Haider, N. (2009, February). Ontology based application level intrusion detection system by using bayesian filter. In Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on (pp. 1-6). IEEE.

Rosa, T. M., Santin, A. O., and Malucelli, A. (2011). Uma Ontologia para Mitigar XML Injection. In: XI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), p. 1-14.

The Strengths of Combining Code Review with Application Penetration Testing. (2009). Retrieved June 12, 2014, from https://www.owasp.org/index.php/The_Strengths_of_Combining_Code_Review_with_Application_Penetration_Testing

Uto, N., and Melo, S. P. (2009). Vulnerabilidades em Aplicações Web e Mecanismos de Proteção. Minicursos SBSeg.

Weske, M. (2007). Concepts, Languages, Architectures (Vol. 14). Berlin: Springer-Verlag. New York, Inc., Secaucus, NJ, United States.

Whitman, M., and Mattord, H. (2011). Principles of information security (3rd ed). Course Technology Press, Boston, MA, United States.

SpamBands: uma metodologia para identificação de fontes de *spam* agindo de forma orquestrada

Elverton Fazzion[1], Pedro Henrique B. Las-Casas[1], Osvaldo Fonseca[1], Dorgival Guedes[1], Wagner Meira Jr.[1], Cristine Hoepers[2], Klaus Steding-Jessen[2], Marcelo H. P. Chaves[2]

[1] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

[2]CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança
NIC.br - Núcleo de Informação e Coordenação do Ponto BR

{elverton,pedro.lascasas,osvaldo.morais,dorgival,meira}@dcc.ufmg.br

{cristine,jessen,mhp}@cert.br

Abstract. *In 2012, estimates indicated that 68.8% of all e-mail traffic was spam, what suggests this is still a relevant problem. Recently, some works have focused on the analysis of spam's traffic inside the network, analyzing the protocols used and the AS which originate the traffic. However, those works usually do not consider the relationships between the machines used to send spam. Such an analysis could reveal how different machines may be used by a single spammer to spread his messages, helping us to understand their behavior. To that end, this work proposes a methodology to cluster the machines used by spammers based on the concept of spam campaigns. The groups identified were characterized to identify different aspects of the spam dissemination process, which suggest different orchestration strategies being used.*

Resumo. *Em 2012, estimava-se que cerca de 68,8% do tráfego de e-mails era spam, o que indica que este problema ainda é muito relevante. Recentemente, os trabalhos têm focado na análise do tráfego de spam na rede, analisando os protocolos usados e os ASes[1] que originam o tráfego. Entretanto, estes trabalhos normalmente não exploram relacionamentos entre as máquinas utilizadas para envio. Esse tipo de análise pode revelar como diversas máquinas podem ser usadas por um spammer para distribuir suas mensagens, ajudando a explicar seus comportamentos. Nesta direção, este trabalho propõe uma metodologia para o agrupamento de máquinas utilizadas por spammers baseada no conceito de campanhas de spam. Os grupos identificados são então caracterizados para identificar diversas facetas do processo de envio de spam, que sugerem diferentes estratégias de orquestração dessas máquinas.*

1. Introdução

Há muitos conceitos sobre o que é *spam*, porém todos têm uma base comum: um *spam* é uma mensagem de email de caráter não individual e não solicitada, que é disseminada em larga escala pela rede. As motivações daqueles que realizam essa prática, os *spammers*,

[1]Sistemas autônomos.

são diversas, sendo as mais comuns a venda de produtos, a disseminação de *malware* e ataques de *phishing* [Crocker 2006]. Segundo a companhia Pingdom, cerca de 144 bilhões de mensagens de email foram enviados por dia, em 2012, sendo 68,8% delas *spam* [Royal Pingdom 2014]. Isso mostra que recursos para enviar e armazenar 99 bilhões de mensagens, por dia, foram desperdiçados, o que leva a sérios prejuízos financeiros, como revelado em outros trabalhos [Sipior et al. 2004]. Além disto, existe um prejuízo social, onde mensagens legítimas são perdidas por má classificação de filtros de *spam* ou por excesso de tráfego ocasionado por grandes volumes de *spam*[Cormack 2008].

Existem diversas facetas consideradas no combate ao *spam*. Muitos estudos buscam entender o problema do ponto de vista do destinatário e auxiliar na construção de filtros eficazes que descartem mensagens indesejáveis. Outros fazem a análise do comportamento do *spammer* na rede, para entender como o *spam* é disseminado, de onde ele se origina e como ele atravessa a rede sem que os transmissores sejam facilmente identificados. O objetivo, nesse caso, é identificar comportamentos na rede que permitam bloquear as mensagens antes que elas atravessem a rede e consumam recursos para sua filtragem e possível armazenamento [Las-Casas et al. 2013].

Em ambos os casos, fica visível que o combate ao *spam* requer o entendimento de um sistema complexo de ofuscação usado pelo *spammer* em sua atividade. Esse sistema exige uma complexa orquestração de atores e recursos, cuja existência é reconhecida mas que normalmente é invisível para o profissional que se dedica a esse combate. Para se manter oculto, o *spammer* busca disfarçar sua localização na rede, seja enviando suas mensagens a partir de múltiplas origens, como máquinas infectadas que se organizam em *botnets*, ou usando servidores especializados que podem por sua vez se aproveitar de máquinas mal-configuradas na rede para se ocultar dos destinatários. Além disso, *spammers* também utilizam programas de transmissão que geram diversas mensagens diferentes como versões de um mesmo conteúdo básico, a fim de tentar ludibriar os filtros baseados em conteúdo [Cormack 2008]. Nesse processo, tem importância o conceito de *campanhas de spam*, que são grupos de mensagens que possuem um mesmo objetivo, mas que foram alteradas por métodos de ofuscação para tentar ludibriar filtros [Guerra et al. 2008a].

Este trabalho utiliza uma abordagem que combina aspectos de campanhas com aspectos de comportamento de rede a fim de tentar lançar mais luz sobre esse elemento orquestrador subjacente ao processo de envio de *spam*. Para este fim, utilizamos tanto elementos baseados no conteúdo da mensagem, para permitir a identificação das *campanhas de spam*, quanto elementos do tráfego de rede, para identificar as máquinas originadoras de cada campanha. Com isso, propomos um método capaz de identificar os grupos de máquinas na rede que se encontram em um certo momento sob o controle de um orquestrador oculto, o *spammer*. A esses grupos denominamos *SpamBands*.

Segundo a abordagem adotada neste trabalho, um(a) *SpamBand* é um grupo de máquinas correlacionadas pelo fato de terem enviado mensagens identificadas como pertencentes a um mesmo conjunto de campanhas de *spam*. Utilizando essa estrutura em nossas avaliações, conseguimos mostrar relações importantes como o período de atividade de cada *SpamBand* e a forma como o *spammer* escolhe o protocolo utilizado. Com relação ao período de atividade, mostramos a tendência desses grupos se manterem estáveis ao longo do tempo, podendo se estender por diversas campanhas e que a técnica pode identificar, como efeito adicional, possíveis partes de redes *botnets*. Quando consideramos a

forma como as mensagens são enviadas, observamos que, *em geral*, *SpamBands* utilizam apenas *proxies* (HTTP ou SOCKS) ou apenas *mail relays* abertos (SMTP) em seus envios, apesar de algumas *SpamBands* apresentarem um comportamento híbrido, utilizando os dois tipos de protocolos.

A definição de *SpamBand* pode facilitar a identificação de *botnets* e outras infraestruturas de distribuição utilizadas pelos *spammers*. Com isso, ações podem ser desenvolvidas para impedir a ação das máquinas envolvidas, removendo-as da rede ou procedendo à remoção de qualquer *malware* nelas instalado. Além disso, pela identificação dos grupos pode se tornar mais eficaz o uso de *blacklists* no bloqueio ao *spam*: se uma máquina é identificada como fazendo parte de um grupo que contém elementos já incluídos em uma lista negra, essa nova máquina também pode ser automaticamente adicionada àquela lista.

2. Metodologia de identificação de *SpamBands*

O conceito de *SpamBands* foi desenvolvido durante a análise dos dados de *spam* coletados em diversos pontos da Internet, onde percebemos que várias origens surgiam na análise do spam observado em diferentes pontos da rede. Nesta seção detalhamos a metodologia proposta para a identificação das *SpamBands* e um exemplo real de aplicação que ressalta alguns elementos importantes da proposta.

Como mencionado, a base do conceito de *SpamBands* é a premissa de que máquinas que enviam mensagens pertencentes às mesmas campanhas são controladas por um mesmo agente orquestrador, estando, assim, relacionadas a uma mesma origem. A relação entre máquinas e campanhas pode ser modelada como um grafo G, onde as máquinas são vértices e há uma aresta entre duas máquinas se elas enviaram mensagens associadas a uma mesma campanha. A figura 1 ilustra a construção desse grafo.

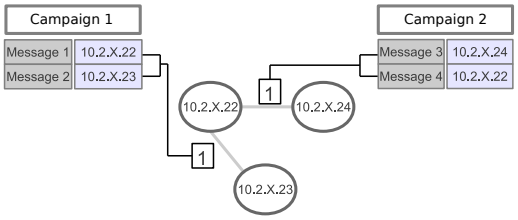


Figura 1. Modelo de grafo para relação entre campanhas e endereços IP

A partir do grafo G, um *SpamBand* pode ser identificado como um sub-grafo denso (diversas origens que compartilham um mesmo conjunto de campanhas). A identificação desses subgrafos pode ser obtida aplicando-se algoritmos de agrupamento de grafos; entretanto, tais algoritmos tendem a ser bastante complexos e difíceis de calibrar [Almeida et al. 2011]. Com base nas características particulares do problema em questão, adotamos uma estratégia mais simples e interativa, descrita a seguir.

Inicialmente, cada componente conectado de G poderia ser identificado como um *SpamBand*. Entretanto, aspectos práticos exigem que essa definição seja refinada. Por exemplo, quando um endereço IP pode se referir a diferentes máquinas atrás de um mecanismo de NAT: duas máquinas podem estar atuando sob coordenações diferentes, mas serem vistas no resto da rede como um mesmo endereço de origem. Em outros casos, um endereço é visto participando de uma campanha até certo instante do dia e a partir de então passa a participar de outra. Os nós referentes a esses endereços IP aparecem

no grafo como nós de ligação entre sub-grafos mais densos, que na prática se referem a *SpamBands* diferentes.

A forma adotada para identificar esses casos e isolar os *SpamBands* envolvidos foi utilizando-se o conceito de *betweenness*, que mede o grau de centralidade de nós em um grafo. Essa métrica quantifica o número de caminhos mínimos entre todos os pares de nós no grafo que passam por um vértice em questão. A premissa é que, se alguns vértices possuem um valor de *betweenness* muito elevado em relação ao que seria esperado para um grafo fortemente conectado, existe uma chance maior desses vértices conectarem dois sub-grafos internamente mais densos. Assim, se removemos esses vértices, acentuamos a separação entre os sub-grafos densos desejados.

A determinação de *SpamBands* é então apresentada no algoritmo 1, que recebe três parâmetros de entrada: o grafo (**G**), o limiar de *betweenness* mínimo a ser considerado (**limiar_bt**) e o número máximo de endereços IP (vértices) que podem ser removidos para dividir um componente (**limiar_ips**). O primeiro passo determina os componentes conectados de G, que constituem uma primeira aproximação dos *SpamBands*. A seguir, identificamos sub-grafos densos em cada componente conectado removendo nós com *betweenness* acima de **limiar_bt**, respeitando o limite **limiar_ips**, que define o tamanho mínimo de um sub-grafo denso, para evitar a geração de conjuntos muito pequenos. O algoritmo retorna o conjunto S que contém todos os *SpamBands*.

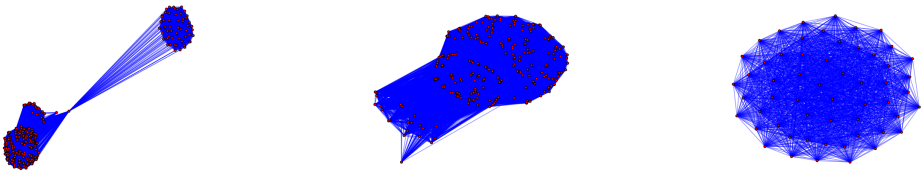
Algorithm 1: *SpamBands* (Grafo G, Real limiar_bt, Real limiar_ips)

```
S = ∅;
C=G.ComponentesConectados(); ;
for comp em C do
    ips_a_remover = ∅ ;
    for ip em comp do
        if ip.Betweenness() > limiar_bt*comp.MaiorBetweenness() then
            ips_a_remover.Adiciona(ip);
        end
    end
    if ips_a_remover.Tamanho() > limiar_ips*comp.Numvertices() then
        S += comp;
    end
    else
        S += comp.RemoveVertices(ips_a_remover);
    end
end
retorna S;
```

Por exemplo, a figura 2(a) mostra um dos componentes conectados com maior número de máquinas observados em um dos dias da nossa análise[2]. Claramente, podemos verificar que há pelo menos dois grupos praticamente disjuntos de nós, unidos por um nó que aparece entre eles. Aplicando o algoritmo 1 naquele componente conectado, isolamos os dois *SpamBands* relativos aos grupos mais densos, mostrado nas figuras 2(b) e 2(c).

Analisamos os *SpamBands* revelados através do componente conectado da fig. 2(a). O *SpamBand* da fig. 2(b) está distribuído em quatro ASes (17816, 17623, 4837 e 17430). Por outro lado, apesar do *SpamBand* da fig. 2(c) estar localizado no mesmo *Country Code*, seu único AS (4134) difere de todos os outros ASes do *SpamBand* da fig. 2(b), o

[2]O dados são provenientes do *honeypot BR-01* do dia 22/07/2013, que faz parte de nosso conjunto de treino.



(a) *SpamBand* original (b) *SpamBand* 1 (c) *SpamBand* 2

Figura 2. *SpamBands*: componente original e aqueles revelados a partir da aplicação do algoritmo 1

que traz uma forte diferença e sugere que esses *SpamBands* identificam *botnets* distintas.

3. Coleta de dados

Os dados utilizados na análise foram coletados utilizando-se oito *honeypots* de baixa interatividade instalados em diferentes partes do mundo: Austrália (AU-01), Áustria (AT-01), Brasil (BR-01 e BR-02), Equador (EC-01), Holanda (NL-01), Taiwan (TW-01) e Uruguai (UY-01). A distribuição desses *honeypots* teve por objetivo capturar dados de diferentes pontos da Internet, a fim de obter uma visão mais global do spam que viaja pela rede.

Todos os *honeypots* foram desenvolvidos de modo a simular computadores com *proxies* HTTP e SOCKS e *mail relays* SMTP abertos, que frequentemente são abusados para o envio de spam. Quando uma máquina se conecta à porta 25 de um dos *honeypots*, ela tem a impressão de estar interagindo com um servidor SMTP operando como um *open relay*, que repassa mensagens de correio para outros servidores. Já uma máquina que se conecta a um *honeypot* através dos protocolos HTTP ou SOCKS, é levada a crer que é capaz de estabelecer conexões para outros servidores SMTP na rede. Toda a interação do atacante com o suposto servidor de correio é registrada e as mensagens de spam são armazenadas localmente — nenhuma mensagem de spam é realmente entregue ao seu destino, exceto mensagens classificadas como mensagens de teste, segundo regras pré-definidas[3]. Periodicamente, ao longo de cada dia, todo o spam armazenado nos *honeypots* é copiado para os servidores centrais do projeto.

O período de coleta usado nesta análise foi de 07/10/2013 a 25/10/2013, totalizando 19 dias consecutivos. A tabela 1 oferece uma visão geral dos dados coletados.

Tabela 1. Visão geral da base

	HTTP(%)	**SMTP (%)**	**SOCKS (%)**	**Total**
Mensagens (milhões)	76,25 (33,7)	32,82 (14,5)	116,58 (51,8)	225,66
Endereços IP	11135 (29,3)	26313 (69,4)	4372 (11,5)	37895
Prefixos de rede	40 (1,5)	2218 (87,7)	342 (13,5)	2529
Sistemas Autônomos (AS)	11 (1,6)	591 (89,0)	125 (18,8)	664
***Country Codes* (CC)**	6 (6,4)	92 (98,9)	31 (33,3)	93
Volume de Tráfego (GB)	211,18 (28,6)	160,74 (21,7)	365,97 (49,7)	737,90

Cerca de 225 milhões de mensagens foram coletadas, provenientes de endereços IP associados a 93 *country codes* distintos. Apesar do protocolo SOCKS ser o responsável pela maior parte do tráfego, representando 51,8% das mensagens enviadas, o número de endereços IP que utilizam o protocolo SMTP é maior, com 69,4% do total, mesmo enviando um número inferior de mensagens.

[3]Presença de texto específico no Subject e/ou corpo da mensagem, rate limiting, etc.

Tabela 2. Mensagens e IPs por *honeypot*

	AT-01	AU-01	BR-01	BR-02	EC-01	NL-01	TW-01	UY-01
Mensagens (milhões)	25,27	6,51	13,89	38,64	16,57	57,52	53,92	13,33
Endereços IP	10438	19420	26762	11261	25494	11053	11145	10138
ASes	330	330	473	142	274	130	122	327

A tabela 2 mostra o número de IPs, o número de mensagens e o número de ASes observados em cada *honeypots*. É importante notar que existe uma sobreposição de IPs entre *honeypots*, que indica que grupos de disseminação de spam estão atuando em mais de um coletor. Este fato será detalhado posteriormente.

4. Resultados

Nesta seção, apresentamos os principais resultados obtidos utilizando a técnica descrita na seção 2. Inicialmente, na subseção 4.1, fazemos um estudo de caso detalhado de forma a mostrar diferentes tipos de *SpamBands* e como estes atuam.

Na subseção 4.2, damos uma visão geral do comportamento dos *SpamBands* encontrados nos *honeypots* e uma possível orquestração de máquinas. Ainda mais, mostramos, por meio de um exemplo, que existem *SpamBands* atuando em diferentes *honeypots*, reforçando a existência de uma orquestração e a eficácia da técnica.

A subseção 4.3 mostra um resultado imediato, obtido através do estudo dos *SpamBands*, no aprimoramento de *blacklists*. Por último, na seção 4.4, apresentamos um estudo temporal dos *SpamBands* com resultados interessantes, realçando o quão valiosa a técnica exposta neste artigo pode ser no estudo dos *spammers* nessa dimensão.

4.1. Estudo de caso

Nesta seção, detalhamos os *SpamBands* descobertos nos dados do exemplo ao final da seção 2. Todos os 7 *SpamBands* podem ser vistos na tabela 3.

Tabela 3. *SpamBands* descobertos no *honeypot BR-01* do exemplo ao final da seção 2

	Msg	IPs	ASes	CC (Top)	SMTP (%)	SOCKS (%)	HTTP (%)	XBL	PBL	Número de horas ativo
***SpamBand* 1**	48.244	971	1	1 (TW)	100	0	0	55	971	24
***SpamBand* 2**	475.971	910	198	52 (CN)	100	0	0	636	597	24
***SpamBand* 3**	2.711	303	4	1 (CN)	100	0	0	224	257	18
***SpamBand* 4**	1.795	56	1	1 (CN)	0	100	0	1	53	23
***SpamBand* 5**	35.389	200	96	26 (BR)	0	100	0	0	56	24
***SpamBand* 6**	28.680	5	1	1 (TW)	0	100	0	0	5	24
***SpamBand* 7**	16.679	3	1	1 (TW)	0	100	0	0	3	24

Entre os *SpamBands* da tabela, podemos identificar três grupos. O primeiro é composto pelos *SpamBands* 6 e 7. Estes *SpamBands* utilizam o procoloco SOCKS e mandam muitas mensagens em relação ao número pequeno de endereços IP que possuem, indicando o uso de servidores dedicados para a disseminação de spam.

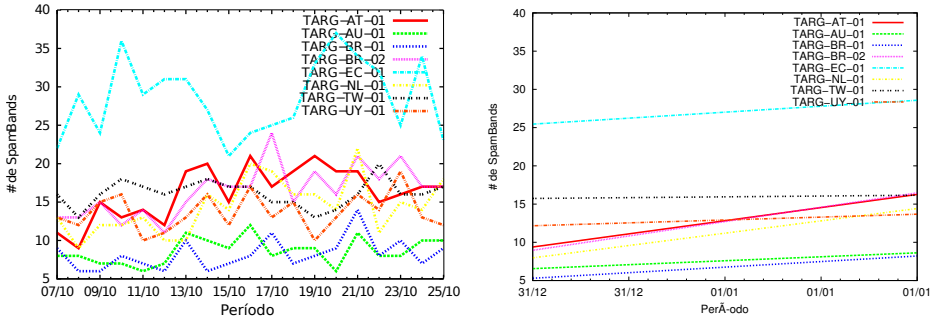
O segundo grupo é formado pelos *SpamBands* 1 e 2, que enviam um número de mensagens muito maior que os demais. Os dois possuem um alto número de endereços IP que estão distribuídos, no *SpamBand* 1, em um AS (3462) do tipo ISP (*Internet Service Provider*) e, no *SpamBand* 2, em 198 ASes. Além disso, grande parte dos endereços IP estão na XBL, o que sugere que estes *SpamBands* podem fazer parte de grandes *botnets* que estão ao redor do mundo e que mandam muito spam.

O terceiro grupo, formado apenas pelo *SpamBand* 5, possui características muito similares aos *SpamBands* 1 e 2, mas o protocolo utilizado é SOCKS. Quatro dos cinco

ASes que mais enviaram mensagens neste *SpamBand* são ASes de *hosting*. Isso leva a crer na possibilidade de que algum grupo de disseminação contratou diversos servidores dedicados para enviar suas mensagens.

O *SpamBand* 3 tem características muito similares ao segundo grupo. Entretanto, ele envia um baixo número de mensagens de *spam* e está concentrado em poucos ASes que estão localizados no mesmo *Country Code*. Isso sugere que este *SpamBand* faz parte de uma pequena *botnet*. Já o *SpamBand* 4 traz indícios de que um único serviço de *hosting* está enviando campanhas de spam pela rede.

4.2. Visão geral dos *SpamBands*



(a) Número de *SpamBands* por *honeypots* durante o período

(b) Regressão linear do comportamento observado na figura 3(a)

Figura 3. Distribuição dos *SpamBands* no período

A técnica aplicada ao longo de 19 dias gerou um total de 2306 *SpamBands*. A figura 3(a) mostra a distribuição desses *SpamBands* ao longo dos dias. Como observado na tabela 2, os dois *honeypots* que mais possuem endereços IP são o *BR-01* e o *EC-01*. Entretanto, observamos no gráfico da figura 3 que esses dois *honeypots* estão em extremos diferentes no gráfico ao longo dos dias, onde o *honeypot EC-01* é o que mais possui *SpamBands* e o *honeypot BR-01*, o que possui menos. Isso sugere que o *honeypot EC-01* é atacado por mais grupos de disseminação de spam do que o *honeypot BR-01*. O restante dos *honeypots* se mantém bem relacionados, mostrando que eles são atacados por um número parecido de grupos de disseminação de spam. A figura 3(b) mostra uma regressão linear do número de *SpamBands* por dia por cada *honeypot*. A linha de tendência revela retas com inclinação suave reforçando que a variação observada na figura 3(a) tem uma regularidade e representa algum tipo de ofuscação utilizada pelo *spammer*.

Relação entre protocolos

Tabela 4. Relações dos protocolos entre *SpamBands*.

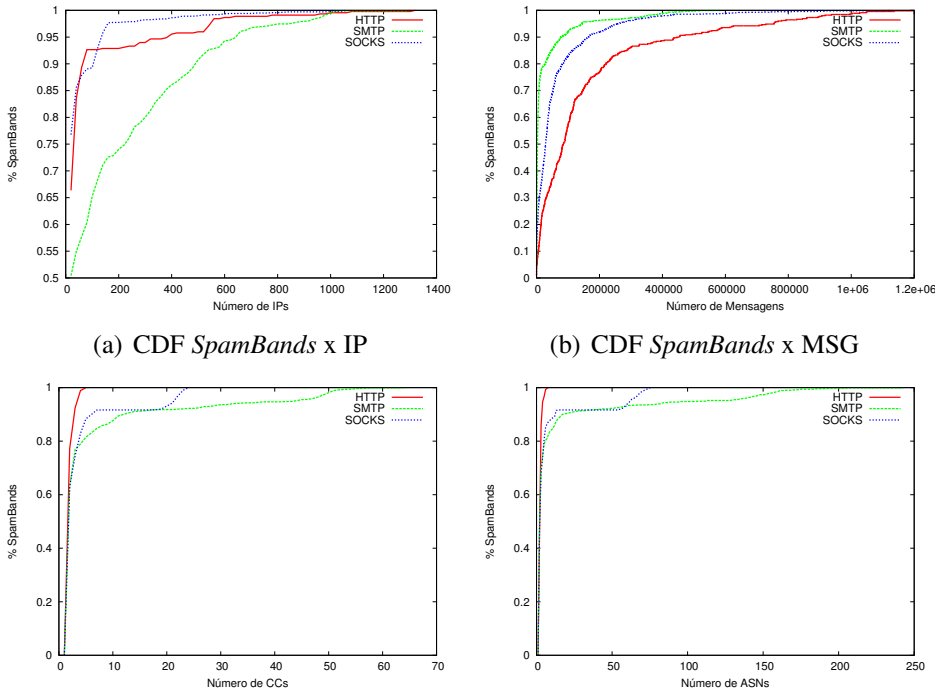
	SpamBands (%)
Somente HTTP	12 (0,52)
Somente SMTP	925 (40,10)
Somente SOCKS	891 (38,62)
Somente HTTP e SMTP	1 (0,05)
Somente HTTP e SOCKS	383 (16,60)
Somente SMTP e SOCKS	42 (1,82)
HTTP e SMTP e SOCKS	53 (2,29)

A tabela 4 mostra a distribuição dos protocolos nos *SpamBands*. Através dessa tabela é possível notar uma relação interessante entre HTTP e SOCKS: entre todos os *SpamBands* que utilizam HTTP, 97,10% também utilizam SOCKS. Como ambos proto-

colos são utilizados para atacar o *honeypot* como *proxy*, isso leva a um forte indício de que o uso desses dois protocolos esteja relacionado com algum tipo de ofuscação, a qual não abordamos mais profundamente neste trabalho.

É possível verificar que muito poucos *SpamBands* utilizam SMTP em conjunto com outro protocolo. Entretanto, existem *SpamBands* que utilizam os protocolos HTTP/SOCKS e SMTP ao mesmo tempo, levando a crer na existência de um grupo de disseminação de spams que utiliza dois ou mais tipos de redes distintas para enviar suas mensagens. Uma possibilidade é o uso tanto de redes *botnets* quanto servidores dedicados para o envio de campanhas de spam. O primeiro tipo de rede tende a utilizar o protocolo SMTP pois o spammer está interessado em apenas repassar suas mensagens, visto que o mesmo já está oculto na rede. No entanto, o uso dos protocolos HTTP e SOCKS, no segundo tipo de rede, indica que o grupo de disseminação de spam utiliza servidores dedicados para o envio de suas mensagens.

Relações entre número de endereços IP, mensagens, CCs e ASes



(a) CDF *SpamBands* x IP

(b) CDF *SpamBands* x MSG

(c) CDF *SpamBands* x CC

(d) CDF *SpamBands* x AS

Figura 4. *SpamBands* em relação ao número de endereços IP, mensagens, CCs e ASes

O gráfico da figura 4(a) mostra que apenas 10% dos *SpamBands* com protocolos SOCKS e HTTP têm mais de 100 endereços IP, o que sugere o uso de servidores para o envio. Entretanto, cerca de 37,5% do total de *SpamBands* que possuem o protocolo SMTP têm mais de 100 endereços IP, o que não surpreende, pois redes *botnets*, em geral, são constituídas por um número maior de endereços IP no envio se comparado com HTTP e SOCKS, além de ter como característica o uso do protocolo SMTP. Entretanto, observando a figura 4(b) verificamos uma inversão: *SpamBands* HTTP e SOCKS tendem a enviar mais mensagens do que *SpamBands* SMTP. Isso sugere que *SpamBands* SMTP, apesar de serem formados por um grande número de endereços IP, enviam poucas

mensagens.

Os gráficos das figuras 4(c) e 4(d) são bastante semelhantes. Aplicando a correlação de Pearson entre o número de *Country codes* e *ASes*, obtemos um coeficiente de 0.95, o que indica que um mesmo *SpamBand* tende a ter comportamento semelhante nos dois gráficos. Dessa forma, a análise para o gráfico 4(c) espelha-se no gráfico 4(d).

O gráfico da figura 4(c) sugere que os *SpamBands* que mais estão espalhados pelos países são SMTP, o que mostra uma característica típica de *botnets*. Todavia, cerca de 85% dos *SpamBands* que utilizam o protocolo SMTP contém endereços IP vindos de menos de 10 CCs, o que indica pequenas *botnets*, similar ao *SpamBand* 3 da tabela 3. Por outro lado, todos os *SpamBands* que possuem HTTP e cerca de 90% que possuem SOCKS têm endereços IP de, no máximo, 5 *Country Codes*, indicando grupos de disseminação que utilizam servidores para o envio de suas mensagens. Entretanto, alguns *SpamBands* que usam o protocolo SOCKS (cerca de 10%) chegam a ter mais de 5 *Country Codes*, indicando um comportamento similar ao *SpamBand* 5 da tabela 3.

Interseção de *SpamBands* entre *honeypots*

Como visto na tabela 2, existe uma recorrência de máquinas entre os *honeypots*. Isso leva a crer que um *SpamBand* pode também participar de outros *honeypots*. Para ilustrar essa reincidência, utilizamos os *SpamBands* do *honeypot BR-01* como referência de comparação com *SpamBands* de outros *honeypots*. A figura 5 apresenta essa visão.

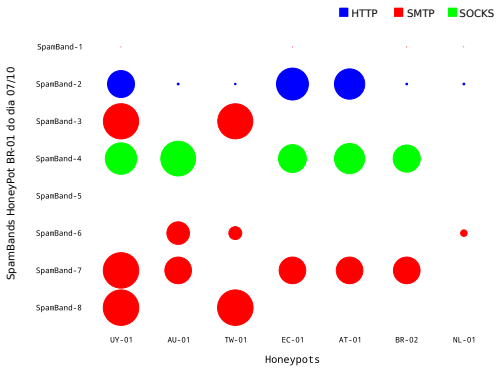


Figura 5. Interseção dos *SpamBands* do *honeypot BR-01* com *SpamBands* de outros *honeypots*

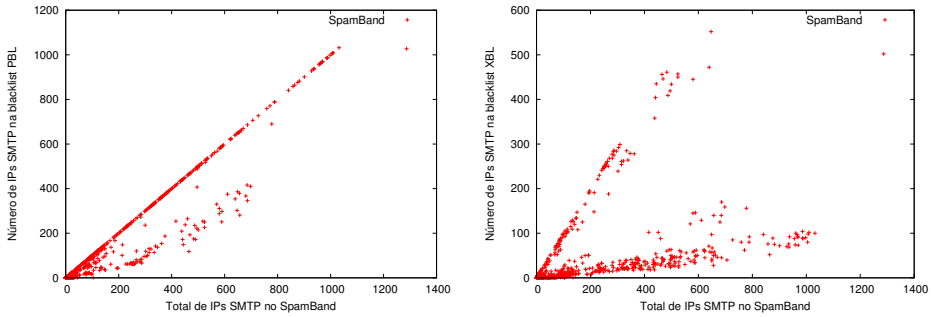
Analisando a figura, é possível verificar que se máquinas de algum *SpamBand* aparecem em outros *honeypots*, elas tendem a estar no mesmo *SpamBand*. O *SpamBand* 5 do *honeypot BR-01*, não possui máquinas em outros *honeypots*, o que indica que esse grupo de máquinas têm visão apenas do *honeypot* usado como referência. Esse fato se assemelha com o *SpamBand* 1, que é o maior em número de máquinas. Entretanto, esse *SpamBand* possui uma única máquina nos *honeypots UY-01, EC-01, BR-02 e NL-01*. Isto leva a crer que o *SpamBand* possui conhecimento dos *honeypots* citados mas, por algum motivo desconhecido, está utilizando apenas o *honeypot BR-01*.

Averiguando os *SpamBands* 2, 3, 4, 6, 7 e 8, vê-se que estes grupos conseguem alcançar outros *honeypots*. Além disso, eles não têm recorrência nos mesmos *honeypots*, o que reforça a hipótese de estes grupos serem independentes. Outro fato importante é que esses *SpamBands* também não utilizam todas as máquinas em todos os *honeypots*. Essa evidência leva a crer que existe algum tipo de distribuição de atividades desses *Spam-*

Bands na rede.

4.3. Relação entre *SpamBands* e *blacklists*

A tabela 5 fornece a relação entre o número de IPs dos *SpamBands* que estão em *blacklists* e o número de IPs que o *SpamBand* possui em cada protocolo. Observamos uma correlação muito forte no número de IPs na *PBL* [4] e o número de IPs do protocolo SMTP nos *SpamBands*. Conforme observado na seção 4.2, 90% dos *SpamBands* possuem somente o protocolo SMTP, o que leva a um forte indício desses *SpamBands* serem partes de *botnets*. Pela figura 6(a) observamos que a *PBL* captura grande parte desses IPs que estão nos *SpamBands* e que possivelmente fazem parte de *botnets*. Por outro lado, os protocolos HTTP e SOCKS possuem uma correlação fraca, o que era esperado visto que *SpamBands* desse tipo tendem a enviar suas mensagens de serviços de *hosting*.



(a) Distribuição de IPs SMTP por IPs SMTP na XBL nos *SpamBands*

(b) Distribuição de IPs SMTP por IPs SMTP na PBL nos *SpamBands*

Figura 6. Relação entre o protocolo SMTP e as *blacklists PBL* e *XBL* nos *SpamBands*

Tabela 5. Coeficiente de determinação entre protocolos dos *SpamBands* e *Blacklists*.

	PBL	XBL
HTTP	0.38	0.11
SMTP	0.86	0.55
SOCKS	0.35	0.08

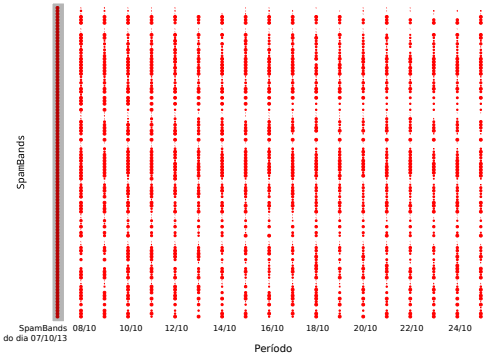
Em relação a *XBL* [5] ,observamos uma correlação moderada, o que não é esperado visto que diversas máquinas que estão em *botnets* estão infectadas por algum tipo de *malware*. Analisando o gráfico da figura 6(b) observamos o porquê da relação ter sido moderada: existem dois eixos de tendência. O primeiro é uma relação linear entre o número de IPs na XBL do *SpamBand* e o número de IPs SMTP, que seria esperado: os endereços IP de todos os participantes de uma *botnet* tendem a acabar sendo identificados por *blacklists*. Entretando, o segundo eixo possui uma relação 1:10, o que sugere algum comportamento especial por parte daqueles *SpamBands*. Eles não só conseguem mascarar bem as atividades de suas máquinas na rede do ponto de vista das *blacklists*, mas mantêm uma taxa comum de identificação em tais listas, o que sugere um comportamento planejado. Determinar a razão para tal comportamento, entretanto, exige novas análises e coletas, sendo considerada como trabalho futuro.

[4] **Policy Block List (PBL):** IPs de usuários finais que não deveriam estar disseminando emails com SMTP não autenticados para qualquer servidor de email, exceto quando especificado pelo ISP.

[5] **Spamhaus Exploits List (XBL):** Banco de dados em tempo real de endereços IPs infectados por *exploits*, incluindo *open proxies* (HTTP, socks, AnalogX, etc), *worms/virus* com *spam engines* e outros tipos de *trojan-horse explois*

4.4. Relação Temporal

Nesta seção, procuramos entender o comportamento dos *SpamBands* do primeiro dia do período avaliado (07/10/2013) em outros dias. O método utilizado para verificar a continuidade do *SpamBand* é recuperar o *SpamBand* do mesmo *honeypot*, no dia seguinte, que mais possui IPs em comum com o *SpamBand* do dia de referência. Observe que esta técnica permite que novos IPs apareçam no *SpamBand* ao longo dos dias e que iremos discutir mais adiante. A figura 7 mostra que existe uma tendência dos *SpamBands* permanecerem ao longo do tempo. O tamanho dos pontos do gráfico indicam quantos IPs permaneceram em relação ao dia de referência.



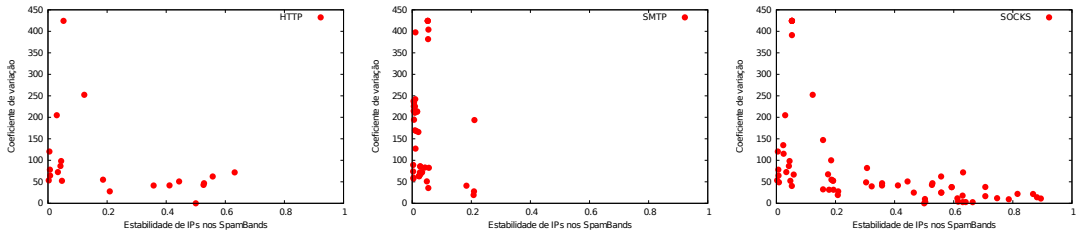
(a) Comportamento dos *SpamBands* do dia 07/10/2013 ao longo dos dias

Figura 7. Comportamento geral dos *SpamBands* ao longo dos dias

Pode-se notar pelo gráfico da figura 7 que os *SpamBands* mudam constantemente seu tamanho ao longo dos dias. Para uma visão geral do comportamento temporal dos *SpamBands* por protocolo, procuramos observar dois quesitos: a variação do tamanho e a estabilidade dos IPs que participam do *SpamBand* no período avaliado. O primeiro quesito é calculado através do coeficiente de variação e o segundo, dividindo a média de IPs pelo número total de IPs distintos que apareceram no período. A figura 8 mostra uma relação global entre protocolos, estabilidade e variação dos *SpamBands*. Pelas figuras 8(a) e 8(c), observamos que os *SpamBands* HTTP e SOCKS tendem a manter seu tamanho e possuir maior estabilidade. Isso reforça, mais uma vez, que os *SpamBands* baseados nesses protocolos utilizam serviços de *hosting* para enviar suas mensagens. Por outro lado, vemos um comportamento diferenciado do protocolo SMTP na figura 8(b), que indica que esses *SpamBands* são bem menos estáveis que os dos protocolos HTTP e SOCKS e possuem maiores variações no tamanho, o que indica uma dinamicidade nesses *SpamBands*.

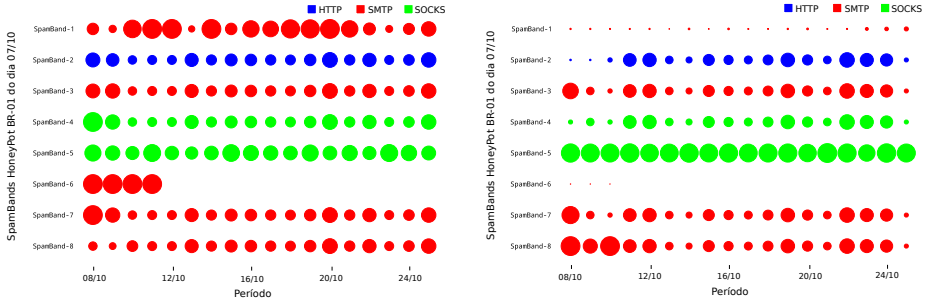
Exemplo de relação temporal entre campanhas e IPs nos *SpamBands*

Para ilustrar o comportamento das campanhas em relação a mudança dos IPs nos *SpamBands*, realizamos o mesmo método aplicado anteriormente para identificar *SpamBands* semelhantes, entre dias, através de IPs. Entretanto, utilizamos campanhas ao invés de IPs. Nos dois gráficos da figura 9, mostramos uma relação entre os IPs e campanhas nos *SpamBands*. A figura 9 mostra o experimento realizado para o *honeypot BR-01*. Como podemos observar na figura, existe uma relação entre o comportamento dos grupos de IPs e campanhas. O *SpamBand* 6 do *honeypot BR-01* no dia 07/10/2013 desaparece



(a) Variação x Estabilidade HTTP (b) Variação x Estabilidade SMTP (c) Variação x Estabilidade SOCKS

Figura 8. Estabilidade e desvio padrão relativo de IPs nos *SpamBands* , ao longo do dia, por protocolo



(a) Comportamento dos IPs dos *SpamBands* do *honeypot BR-01* do dia 07/10 ao longo do período analisado.

(b) Comportamento das campanhas nos *SpamBands* do *honeypot BR-01* do dia 07/10 ao longo do período analisado.

Figura 9. Exemplo de relação entre campanha e IPs nos *SpamBands*

completamente no dia 12/10/2013, como mostra a figura 9(a). Todos os 47 IPs deste *SpamBand* são SMTP e fazem parte de apenas um *country code* (CN) e um AS (4134) do tipo DSL (4134), o que indica um *SpamBand* que faz parte de uma pequena *botnet*. Como os IPs do tipo DSL tendem a ser dinâmicos, é possível essas máquinas finais tenham mudado seu endereço IP durante os dias. Entretanto, a possibilidade delas terem saído da *botnet* é maior visto que as campanhas que elas suportavam também desapareceram.

O *SpamBand* 5 retrata um grupo puramente SOCKS, onde IPs estão distribuídos em 23 *country codes* e 72 ASes. Esse grupo é semelhante ao *SpamBand* 5 do estudo de caso da seção 4.1, que possivelmente contratou diversos serviços de *hosting* para o envio das campanhas. Observe que o grupo de máquinas permanece similar ao longo do tempo e existe uma periodicidade no grupo de campanhas, mostrando que este grupo de máquinas enviam um mesmo grupo de campanhas alternadas ao longo dos dias avaliados.

Um outro comportamento interessante que podemos observar é sobre os *SpamBands* 2,3,4,7 e 8. No dia 11/10, estes *SpamBands* se unem, formando um único *SpamBand* e, por isso, o comportamento temporal desses 5 *SpamBands* nos dois gráficos da figura 9 são iguais. Este fato indica que estes *SpamBands* podem estar oferecendo serviços que são contratados por grupos de disseminação e que, em algum momento, um grupo contratou estes serviços para enviarem as mesmas campanhas.

5. Trabalhos Relacionados

Alguns autores já focaram no comportamento dos *spammers* de formas que tiveram impacto sobre este trabalho.

Guerra et al. apresentam uma técnica que utiliza uma estrutura de mineração de dados denominada FPTree para agrupar mensagens de spam [Guerra et al. 2008b]. As mensagens assim agrupadas definem o conceito de campanha de spam, como usadas neste trabalho: uma campanha é um conjunto de mensagens que foram enviadas com um mesmo objetivo mas que foram diferenciadas por algum tipo de ofuscação, com a finalidade de não serem captadas por filtros spam.

Ramachandran et al. mostram que o *spammer* alterna as máquinas usadas para envio, de modo ocultar sua origem e contornar diversos filtros de spam na rede [Ramachandran and Feamster 2006]. Esse trabalho sugere que as mensagens de uma mesma campanha podem ser enviadas por diferentes máquinas, o que motiva nosso trabalho para encontrar uma forma de agrupar essas máquinas.

Moreira Moura et al. introduz o conceito de *Bad Neighborhoods*, que são vizinhanças de rede com alta probabilidade de um IP enviar spam [Moreira Moura et al. 2011]. *Fonseca et al.* estende esse conceito e estabelece uma relação direta de vizinhança com Sistemas Autônomos (AS), por esses terem fronteiras bem definidas. Nosso trabalho apresenta uma visão complementar a esses conceitos: ao invés de focarmos diretamente nos locais de origem do *spam*, estamos procurando entender como diferentes origens (máquinas em diferentes pontos da rede) se relacionam para atender às necessidades do *spammer*, o orquestrador por trás de todo o processo.

Por fim, Zhuang *et al.* associa características de spam a *botnets*, que são um meio de envio de mensagens de spam [Zhuang et al. 2008]. Contudo, existe a possibilidade de que vários *spammers* utilizem a mesma *botnet* ou combinações delas, visto que essas redes muitas vezes são alugadas para terceiros [Raywood 2010]. Dessa forma, sem uma identificação das campanhas de spam sendo enviadas, grupos de máquinas utilizadas por diferentes spammers podem ser vistas como uma só entidade, não levando a um bom agrupamento de máquinas.

6. Conclusão e Trabalhos Futuros

Neste trabalho, buscamos entender melhor o comportamento dos *spammers* correlacionando as máquinas utilizadas para envio através das campanhas de spam enviadas. Para realizar essa análise, propusemos o conceito de *SpamBands*, grupos de máquinas que participam das mesmas campanhas e sugerem a existência de um único orquestrador por trás de seu comportamento e desenvolvemos uma metodologia baseada em grafos para identificar esse grupos. Inicialmente, conectamos todas as máquinas que enviam as mesmas campanhas. Os grupos revelados por esta metodologia passam por um processo de refinamento, de forma a separar subgrafos densos, que revelam os *SpamBands*.

Descobrimos que a grande maioria dos *SpamBands* tendem a utilizar apenas o protocolo SMTP ou os protocolos HTTP/SOCKS, o que faz uma distinção entre grupos que utilizam servidores dedicados e redes *botnets* para o envio de spam. Além disso, mostramos que esse conceito permite revelar grupos que não são inteiramente detectados pela *blacklist XBL*, podendo ajudar a inferir outras máquinas que deveriam pertencer à *blacklist*. Ademais, encontramos *SpamBands* que utilizam os dois tipos de protocolos, levando a crer na existência de grupos de disseminação de spam que utilizam tanto servidores dedicados quanto redes *botnets* para enviar mensagens.

Por fim, realizamos ainda um estudo sobre os *SpamBands* ao longo dos dias ava-

liados, revelando que eles se repetem ao longo do tempo. Nesta avaliação, descobrimos que *SpamBands* que utilizam os protocolos HTTP/SOCKS tendem a ser mais estáveis em relação ao número de IPs, o que não acontece com *SpamBands* que utilizam o protocolo SMTP. Como trabalho futuro, pretendemos analisar mais profundamente o comportamento dos *SpamBands* ao longo dos dias, buscando a existência de uma interação entre eles, de forma a entender o comportamento temporal.

Agradecimentos

Este trabalho foi parcialmente financiado por NIC.BR, Fapemig, CAPES, CNPq e InWeb.

Referências

Almeida, H., Guedes, D., Meira, W., and Zaki, M. J. (2011). Is there a best quality metric for graph clusters? In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, pages 44–59, Athens, Greece.

Cormack, G. V. (2008). Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1(4):335–455.

Crocker, D. (2006). Challenges in anti-spam efforts. *The Internet Protocol Journal*, 8(4).

Guerra, P. H. C., Guedes, D., Jr., W. M., Hoepers, C., and Steding-Jessen, K. (2008a). Caracterização de estratégias de disseminação de spams. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBC.

Guerra, P. H. C., Pires, D. E. V., Guedes, D., Wagner Meira, J., Hoepers, C., and Steding-Jessen, K. (2008b). A campaign-based characterization of spamming strategies. In *Proceedings of the 5th Conference on e-mail and anti-spam (CEAS)*, Mountain View, CA.

Las-Casas, P. H. B., Guedes, D., Jr., W. M., Hoepers, C., Steding-Jessen, K., Chaves, M. H. P., Fonseca, O., Fazzion, E., and Moreira, R. E. A. (2013). Análise do tráfego de spam coletado ao redor do mundo. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBC.

Moreira Moura, G. C., Sadre, R., and Pras, A. (2011). Internet bad neighborhoods: the spam case. In Festor, O. and Lupu, E., editors, *7th International Conference on Network and Services Management (CNSM 2011), Paris, France*, pages 1–8, USA. IEEE Communications Society.

Ramachandran, A. and Feamster, N. (2006). Understanding the network-level behavior of spammers. *SIGCOMM Comput. Commun. Rev.*, 36(4):291–302.

Raywood, D. (2010). The botnet market and what you get for your money. *SC Magazine UK*.

Royal Pingdom (Visitado em 2014). The internet 2012 in numbers. Artigo na Web.

Sipior, J. C., Ward, B. T., and Bonner, P. G. (2004). Should spam be on the menu? *Commun. ACM*, 47(6):59–63.

Zhuang, L., Dunagan, J., Simon, D. R., Wang, H. J., Osipkov, I., and Tygar, J. D. (2008). Characterizing botnets from email spam records. In Monrose, F., editor, *LEET*. USENIX Association.

CloudSec - Um Middleware para Compartilhamento de Informações Sigilosas em Nuvens Computacionais

Rick Lopes de Souza, Hylson Vescovi Netto, Lau Cheuk Lung, Ricardo Felipe Custódio

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

{rick.lopes, hylson.vescovi, lau.lung, custodio} @inf.ufsc.br

Abstract. *The need to share and manipulate sensitive data is a challenge for most content providers using the cloud for storage. However, developing applications that guarantee confidentiality in the cloud are complex and require integration of multiple aspects of security and interoperability. To circumvent these challenges, this paper aims to propose a middleware architecture to ensure secure sharing of documents using public cloud providers for data storage and hardware secure modules for cryptographic key management. This work has as main features: the use of identity-based encryption, use of hybrid clouds, simplified management of cryptographic keys, peer-to-peer security assurance and use of cryptographic security modules.*

Resumo. *A necessidade de compartilhar e manipular dados sensíveis é um desafio para grande parte dos provedores de conteúdo que utilizam a nuvem para armazenamento. No entanto, desenvolver aplicações que garamtam sigilo em nuvem é uma tarefa complexa e requer integração de múltiplos aspectos de segurança e interoperabilidade. Para contornar esses desafios, esse trabalho tem como principal objetivo propor uma arquitetura de middleware para garantir o compartilhamento seguro de documentos sigilosos em nuvem utilizando provedores de nuvens públicas para o armazenamento dos dados e módulos de segurança criptográficos para o gerenciamento de chaves criptográficas. Este trabalho tem como principais características: o uso da criptografia baseada em identidade, uso de nuvens híbridas, gerenciamento de chaves criptográficas simplificado, garantia de segurança ponto a ponto e utilização de módulos de segurança criptográficos.*

1. Introdução

A necessidade de garantir o sigilo utilizando provedores de nuvem pública para armazenamento é grande, entretanto, a implementação e integração de serviços para garantir todas as propriedades necessárias de segurança é complexo e pode comprometer o desenvolvimento de um produto, caso não seja bem feito. Propriedades como o correto e seguro gerenciamento das chaves criptográficas são essenciais em sistemas de sigilo, assim como tolerância a falhas [1] e a retirada da necessidade de se confiar inteiramente nos provedores de nuvens públicas.

Grande parte das empresas já possui um software para gerenciamento de documentos, assim como um controle de acesso já estabelecido. Entretanto, esses softwares

[1]Considera-se que uma falha poderá ocasionar um erro no sistema que levará a um defeito no funcionamento.

carecem de mecanismos de segurança para proteger os dados dos usuários e corporações que os utilizam. O desenvolvimento de mecanismos de segurança não é uma tarefa trivial e quando feita por empresas não especializadas, pode levar a falhas de segurança, expondo assim dados sensíveis. Para que isso seja feito de uma maneira mais segura, necessita-se de um software que gerencie os mecanismos de segurança de uma forma transparente e integrando diferentes tipos de comunicações e operações criptográficas.

Problema: A dificuldade de integrar diferentes mecanismos de segurança de forma transparente à uma aplicação que deseja garantir o compartilhamento de informações sigilosas de forma segura e confiável. Dentre os mecanismos de segurança existentes, o gerenciamento de chaves criptográficas é um ponto crítico e deve ser tratado com mais atenção, assim como a tolerância a falha nos sistemas de distribuição de chaves criptográficas e armazenamento de dados em nuvens computacionais.

Motivação: A transparência na integração de serviços de segurança é essencial para outras aplicações que necessitam desses serviços. Grande parte das corporações não possuem conhecimento muito avançado sobre os conceitos de segurança da informação e, com isso, implementações com falhas podem ser colocada no mercado, podendo expor informações sensíveis de usuários e corporações. Dessa forma, necessita-se de soluções que envolvam proteções físicas e lógicas para o gerenciamento das chaves criptográficas dos usuários, fazendo com que seja oferecido segurança, disponibilidade e usabilidade. Além disso, deve-se utilizar mecanismos que garantam a tolerância a falhas e segurança na utilização dos provedores de nuvens públicas para armazenamento, fazendo com que estes não tenham condições de acessar os dados sigilosos dos usuários.

A partir do problema e motivação, conclui-se que são necessárias soluções para proteger de forma física e lógica o gerenciamento de chaves criptográficas dos usuários. Assim como outros mecanismos de segurança como tolerância a falhas para prover segurança, disponibilidade e usabilidade aos usuários. Essas soluções devem ser transparentes para as aplicações dos usuários.

Contribuição: A principal contribuição desta proposta é uma arquitetura de middleware para compartilhar documentos sigilosos por meio de nuvens públicas obtendo-se as principais garantias de segurança necessárias no gerenciamento de chaves criptográficas, fazendo com que a integração desses serviços fique transparente para uma aplicação do usuário que necessita utilizar esses serviços. A arquitetura proposta torna prática a manutenção de usuários e grupos utilizando Criptografia Baseada em Identidade com multi autoridades, segredo compartilhado e *erasure codes*. Este trabalho envolve temas de pesquisa como revogação segura de usuários, correto gerenciamento da privacidade dos documentos para grupos de usuários, chaves privadas geradas por demanda, segurança física e lógica utilizando módulos de segurança criptográficos 3.3, tolerância a falhas e um compartilhamento eficiente de documentos.

Este artigo está organizado da seguinte forma. Na seção 2 são apresentados a fundamentação matemática e os trabalhos relacionados. Na seção 3 é apresentada a arquitetura da solução proposta. A seção 4 apresenta algoritmos detalhados das principais funções do middleware. Na seção 5 são feitas avaliações a respeito da segurança e usabilidade da arquitetura proposta. Por fim, na seção 6 são apresentadas as conclusões do estudo.

2. Fundamentação Matemática e Trabalhos Relacionados

2.1. Geração de Chaves Distribuída

Neste trabalho utiliza-se um mecanismo distribuído de geração de chaves mestras de CBI baseado no protocolo de Joint-Feldman Distributed Key Generator (JF-DKG), o protocolo de geração distribuído modificado e proposto por Aniket [Kate et al. 2012]. O protocolo JF-DKG, o mais simples e eficiente dos geradores distribuídos de chaves, requer um número $n \geq 3t + 1$ de nodos para funcionar corretamente, sendo t a quantidade de nodos que podem falhar. Este trabalho utiliza a técnica de BF-IBE [Boneh and Franklin 2001] devido a sua simplicidade nos protocolos de inicialização e métodos de criptografia. Na inicialização, um Gerador de Chaves Privadas (GCP) gera as chaves privadas (d) dos usuários utilizando suas identidades (ID) e uma chave mestra (s). Este trabalho visa uma geração de chave distribuída (n, t) por meio de um grupo de curvas elípticas G com uma ordem q e um gerador U, onde n é o número total de nodos envolvidos e $t + 1$ nodos honestos são suficientes para gerar corretamente o segredo. Seja F(z) = $a_0 + a_1 z + ... + a_t z^t \in Z_q[z]$ o atual polinômio compartilhado e s = a_0, sendo s o segredo.

O protocolo proposto por Aniket utiliza uma versão melhorada do protocolo de Feldman Verifiable Secret Sharing (Feldman VSS) para gerar de forma distribuída a chave mestre. Este possui um "quadro de avisos" que gera os parâmetros públicos da inicialização do BF-IBE, publica os valores e inicializa os valores de A_k e A_{ik} em zero, para i = 1, ..., n e k = 0 ... t, onde $A_k = a_k U$ e $A_{ik} = a_{ik} U$. A chave mestra é inicializada em zero. Os nodos inicializam o protocolo Feldman VSS. Depois que $t + 1$ nodos terminarem com sucesso o protocolo, as partes são consideradas seguras. Estes nodos são então chamados de nodos qualificados. Aqui, estes nodos são denotados como ∂. O quadro de avisos então computa e transmite os coeficientes A_k (para k = 0... t) para os polinômios compartilhados F(z).U como $A_k = \sum_{P_j \in \partial} A_{ik}$. Depois de verificar os A_k valores, os nodos enviam assinaturas de confirmações para o quadro de avisos. Depois de receber $t + 1$ confirmações, o valor A_k é finalizado. Cada nodo então computa sua parte do segredo como $s_i = \sum_{P_j \in \partial} s_{ji}$.

2.2. Extração da Chave Privada

Para extrair a chave privada de forma distribuída, o usuário deve entrar em contato com os nodos e enviar um ID específico. Após receber o ID, autenticar e autorizar o usuário, os GCP $P_i \in O$ retornam uma parte da chave privada $S_i H(ID)$ por meio de um canal seguro. O símbolo H representa uma função de resumo criptográfico $H : (0, 1)^* \rightarrow G^*$. Depois de receber $t+1$ partes corretas da chave privada, o usuário pode reconstruir a chave privada da seguinte maneira: $D_{id} = \sum_{P_i \in O} \lambda_i s_i H(ID)$, onde o coeficiente de Lagrange é $\lambda_i = \prod_{P_j \in O, j \neq i} \frac{j}{j-i}$.

2.3. Trabalhos Relacionados

Trabalhos recentes ([Itani et al. 2009], [Pearson et al. 2009]) propõem o uso de serviços de privacidade para resolver o problema do armazenamento de documentos sensíveis, assim como outros trabalhos ([Padilha and Pedone 2011], [Singh et al. 2011]) que propõem não cifrar os arquivos, apenas quebrá-los e enviar para diferentes provedores de nuvem. Estes trabalhos tentam contornar os problemas envolvidos no armazenamento de documentos sensíveis na nuvem, entretanto, não conseguem prover as propriedades necessárias

para garantir um compartilhamento seguro. O trabalho de Itani et al. ([Itani et al. 2009]) não provê um esquema de compartilhamento tolerante a falhas. Caso o serviço de privacidade seja interrompido, o cliente não pode cifrar ou decifrar arquivos. A proposta de Padilha e Pedone ([Padilha and Pedone 2011]) usa a técnica de homomorfismo para modificar as partes dos arquivos cifrados por meio de funções aditivas, contudo, as técnicas para prover sigilo utilizando homomorfismo total são teóricas e não existem implementações pragmáticas. Implementações práticas do homomorfismo total para sistemas de sigilo ainda são assuntos abertas de pesquisa, portanto, não são aplicáveis nas atuais circunstâncias das corporações.

Outra linha de trabalho é a Criptografia Baseada em Atributos (CBA), onde cada usuário recebe credenciais de autoridades confiáveis, liberando então o acesso aos dados sensíveis. Alguns trabalhos ([Ruj et al. 2011], [Jung et al. 2013], [Yang et al. 2012a] e [Yang et al. 2012b]) propõem o uso de multi autoridades para gerar as chaves privadas, evitando assim que as autoridades distribuidoras de chaves possuam controle das chaves privadas (*key escrow*). Esses trabalhos também funcionam de maneira distribuída para fornecer os atributos, conseguindo assim suprir as necessidades da confidencialidade das identidades. No entanto, as soluções que envolvem CBA possuem a desvantagem da revogação. Uma vez que uma chave é revogada, o sistema necessita recifrar todos os documentos sensíveis e, desta forma, gerar novas chaves para todos os usuários. Outra peculiaridade em quase todos os trabalhos é que a tolerância a falhas não é considerada. As propostas são baseada em multi autoridades apenas para a extração das chaves privadas e atributos, não obstante, armazenam os arquivos cifrados em apenas um provedor de nuvem. Caso este provedor de nuvem falhe por qualquer razão, o usuário não conseguirá ter acesso a seus dados sensíveis. Outro problema devido a este fato é no caso do comprometimento de uma chave, se o atacante tiver algum tipo de acesso ao provedor de nuvem, ele pode obter o dado integral do documento, pois ele estará cifrado e completo em apenas um local.

O sistema DepSky, de Bessani et al [Bessani et al. 2011], usa conceitos que foram utilizados neste trabalho: criptografia simétrica, segredo compartilhado de Shamir e *erasure optimal code*. Contudo, o artigo não propõe mecanismos necessários para compartilhar as chaves que garantem a integridade das partes dos arquivos sigilosos. O trabalho simplesmente admite que existe um mecanismo para compartilhamento, no entanto, este é exatamente um dos principais desafios para a garantia do sigilo na nuvem utilizando criptografia. Outra fragilidade no artigo é o algoritmo de leitura de um dado que não verifica a integridade do resumo criptográfico das partes com a chave pública. Caso o provedor de nuvem seja malicioso, poderá modificar as partes e prover falsos resumos criptográficos no momento da verificação da integridade, comprometendo então o funcionamento do sistema.

O trabalho de Zhou et al [Zhou et al. 2011] utiliza a técnica de BF-IBE modificada para impor uma criptografia baseada em papéis, possibilitando cifrar um documento para um usuário específico ou para grupos de usuários com um determinado papel. A proposta é eficiente apenas quando não há muitas revogações, caso contrário, apresentará alta complexidade. O trabalho de Zhou não resolve o problema das custódias das chaves (*key escrow*). O administrador do sistema tem acesso às chaves privadas no método de extração. Outro problema é o ponto único de falha, fazendo com que caso o administrador

seja comprometido, parte do sistema de revogação também seja comprometida.

Baseado em pesquisas anteriores, este artigo identificou os principais desafios para compartilhar documentos sensíveis de uma forma segura. Foi proposta uma arquitetura de middleware para tentar superar todos os desafios envolvendo o uso da CBI para prover as seguintes propriedades: revogação segura de usuários, geração de chaves por demanda, gerenciamento seguro dos grupos, tolerância a falhas, economia no armazenamento e compartilhamento eficiente de documentos.

3. Arquitetura do CloudSec

3.1. Premissas

Este trabalho tem como principal foco propor uma arquitetura de middleware para prover um compartilhamento seguro de documentos sigilosos:

- O controle de acesso é replicado em cada servidor de nuvem ou módulo de segurança criptográfico 3.3;
- Não existe concorrência para realizar alterações no controle de acesso;
- Os provedores de nuvem são semi-confiáveis (Estes irão se comportar corretamente perante as requisições dos usuários, mas são curiosos para ver os dados armazenados);
- Existe um versionamento dos documentos que são escritos na nuvem;
- O middleware será disponibilizado como uma biblioteca que deverá ser integrada a aplicações já existentes.

3.2. Arquitetura

A arquitetura possui três componentes principais: Os provedores de nuvem pública para armazenamento, usuários finais que desejam compartilhar documentos sigilosos e gerenciadores de chaves criptográficas. A figura 1 ilustra a arquitetura do middleware. Cada componente é melhor detalhado a seguir:

- **Gerenciadores de Chaves Criptográficas:** Estarão instalados internamente em diferentes MSCs em nuvens privadas, com ambientes controlados fisicamente e com acesso à Internet. Deve-se empregar uma distribuição geográfica, fazendo com que os MSCs possam ser acessados por meio da Internet mas que estejam geodistribuídos o suficiente para não serem comprometidos facilmente de forma física. Esse gerenciamento utilizará conceitos de criptografia baseada em identidade em conjunto com um rígido controle de acesso.
- **Provedor de Nuvem Pública:** Os servidores de armazenamento precisam estar hospedados em diferentes provedores de nuvem. As principais características necessárias são: a distribuição do controle de acesso por meio de replicação de estados e o funcionamento distribuído nos nodos dos servidores de aplicação. Os provedores de nuvem são responsáveis pelo armazenamento dos documentos e pelo controle de acesso aos dados.
- **Usuário:** O lado do usuário é responsável por editar, cifrar e decifrar arquivos. Os usuários possuem uma aplicação local que interage com o middleware para realizar as funções criptográficas e funções de compartilhamento dos arquivos sigilosos. Os mesmos definem quem serão os custodiantes dos documentos sensíveis cifrados. Esses custodiantes são delimitados por regras de acesso específicas de cada aplicação.

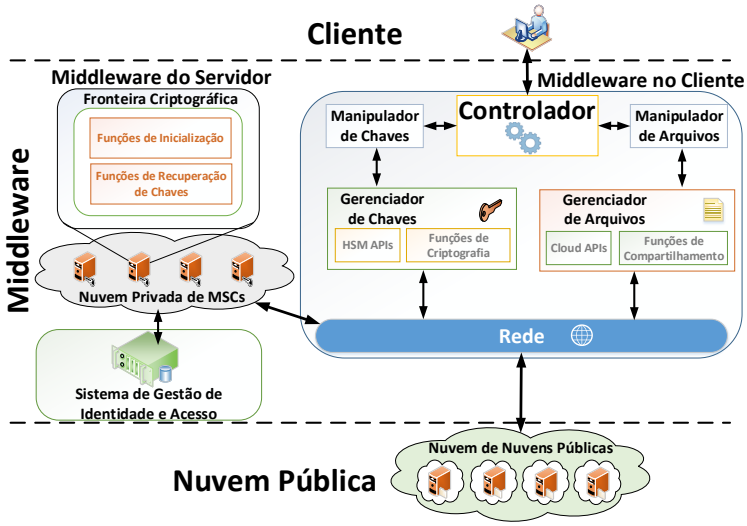


Figura 1. Arquitetura do Middleware de Privacidade em Nuvem.

A arquitetura do middleware possui diferentes módulos para a integração de serviços para aplicações que desejam proporcionar o sigilo no compartilhamento de documentos sensíveis. Os módulos são os seguintes:

- **Controlador:** Esse módulo será o responsável por disponibilizar uma interface de programação de aplicativos (*Application Programming Interface* - API) para o usuário, a fim de que seja feita uma integração entre os serviços de armazenamento em nuvem, gerenciamento de chaves nos módulos de segurança criptográficos e serviços de criptografia.
- **Manipulador de Chaves:** Módulo responsável pelas chamadas que envolvem criptografia, como a criação de chaves simétricas e a construção das chaves assimétricas baseada em identidade, assim como funções que envolve processos de transformação dos dados em claro para dados criptografados. Esse módulo também é responsável por criar localmente chaves simétricas para sigilo e chaves assimétricas baseadas em identidade por meio do módulo HSM API.
- **Manipulador de Arquivos:** Módulo responsável pelas chamadas que envolvem o armazenamento e recuperação de arquivos sigilosos e que estão cifrados e assinados pelo Manipulador de Chaves. Disponibilizará funções para separar e juntar arquivos e chaves criptográficas envolvidas no processo de sigilo. Disponibilizará também as diferentes APIs utilizadas por diferentes provedores de nuvens, com o objetivo de estabelecer a correta comunicação com todos para o compartilhamento de arquivos.

Essa arquitetura prevê que toda a infraestrutura de provedores de nuvens públicas para armazenamento e os módulos de segurança criptográficos estejam previamente configurados e com seus controles de acesso pré-definidos. O middleware ficará encarregado de fazer toda a integração entre os processos de autenticação, armazenamento e sigilo, fazendo com que todo o procedimento de compartilhamento de documentos sigilosos seja o mais seguro e transparente possível para o usuário.

A arquitetura utiliza internamente o esquema de segredo compartilhado para integrar a confidencialidade com a disponibilidade. Uma vez que todas as chaves simétricas

cifradas são quebradas em N pedaços (sendo N o número total de provedores de nuvem), o usuário necessitará de um número mínimo de M partes (M é um número pré-estabelecido no momento da inicialização do sistema) para reconstruir a chave cifrada. De fato, este trabalho reutiliza o controle de acesso da aplicação para controlar quais leitores estarão habilitados para acessar o dado armazenado. Este trabalho também utiliza o mecanismo de *information-optimal erasure code* [Plank et al. 2008], possibilitando uma economia nos provedores de nuvem ao armazenar os arquivos de tal forma que cada parte é reduzida por um fator de $\frac{n}{f+1}$, considerando f o número de servidores com falhas.

Para garantir propriedades de segurança no gerenciamento das chaves criptográficas, utiliza-se o esquema BF-IBE para cifrar chaves simétrica construindo um identificador ID específico contendo as seguintes informações: Nome do Documento, Grupo de Custodiantes e Versão do Documento. Este identificador específico ID é utilizado devido a uma série de características que são necessárias para compartilhar documentos sensíveis. O Nome do Documento no ID é utilizado para que cada documento cifrado e armazenado na nuvem possua uma chave diferente. O Grupo de Custodiantes é para limitar o controle de acesso ao documento e como estaremos reutilizando o controle de acesso do sistema, este grupo de custodiantes será utilizado para autenticar e liberar o acesso aos documentos sigilosos. Para cada grupo diferente de custodiantes existirá uma chave diferente. O Número de Versão juntamente com os outros elementos são utilizados para controlar o acesso de diferentes versões dos documentos e, desta forma, garantir a segurança na entrada e saída de membros do grupo. A técnica de Assinatura Hess [Hess 2003] é utilizada para assinar as partes cifradas das chaves e documentos para garantir a integridade. Neste trabalho é utilizado o mesmo par de chaves para cifrar e assinar, facilitando assim o gerenciamento e compartilhamento das chaves.

3.3. Módulo de Segurança Criptográfico

O módulo de segurança criptográfico (MSC) [2] tem como principal função o correto gerenciamento das chaves criptográficas, protegendo-as de forma física e lógica. Para isso, os MSCs serão previamente configurados com um firmware contendo os aplicativos necessários para o seu correto funcionamento, incluindo o aplicativo gerenciador de chaves criptográficas. Com esse firmware, o MSC disponibilizará o aplicativo de gerenciamento das chaves privadas baseadas em identidade utilizando APIs próprias para comunicação com os usuários. Essa API deve ser simples o suficiente para não prejudicar a segurança do MSC, mas deve prover todas as funções necessárias para o correto gerenciamento das chaves criptográficas.

Os MSCs devem inicialmente ser configurados para compartilharem partes de uma chave mestra. Para isso, o aplicativo responsável nos MSCs deverá executar os passos citados na seção 2.1. Com isso, todos os MSCs envolvidos no gerenciamento das chaves criptográficas baseadas em identidade compartilharão uma parte da chave mestra que gerará as chaves privadas dos usuários do middleware.

3.4. Implementação

Com o CloudSec, pode-se compartilhar e recuperar dados sigilosos entre grupos de usuários utilizando nuvens híbridas por meio de funcionalidades específicas. Essas

[2]Exemplo de um MSC Nacional: ASI-HSM `http://www.kryptus.com/#!asi-hsm/c11e6`

funções tem como principal objetivo prover ao usuário funções reutilizáveis e configuráveis para o desenvolvimento mais rápido de sistemas que necessitem de compartilhamento de documentos sigilosos. A figura 2 ilustra o diagrama com as principais classes referentes a essa arquitetura. As classes *Storage Handler* e *Crypto Handler* são as principais classes de controle do middleware e distribuem as requisições conforme forem solicitadas pela classe *Controller*.

Figura 2. Diagrama de Classe ilustrando os principais componentes do CloudSec.

Para a realização dos procedimentos de compartilhamento de documentos sigilosos, o CloudSec provê para as aplicações as funcionalidades de armazenamento e criptografia por meio das seguintes APIs:

- **searchForMetadata**$(dataId)$**:** Para o compartilhamento ou recuperação de arquivos sigilosos, deve-se procurar primeiramente pelos metadados de um determinado arquivo para verificar suas informações. Para isso, deve-se especificar o nome do arquivo que deseja-se compartilhar. Caso o documento exista nos provedores de nuvens públicas, os metadados serão retornados contendo informações a respeito dos grupos que possuem acesso, assim como as versões existentes no sistema de armazenamento. Caso o documento não exista, é retornado ao usuário um metadado nulo.
- **sharePrivateData**$(dataPath, Metadata)$**:** Para que um usuário do sistema possa compartilhar um arquivo, deve-se passar o caminho do mesmo ($dataPath$), assim como um metadado ($Metadata$) contendo informações desse arquivo, como nome, versão e grupo que poderá ter acesso ao mesmo. Antes de realizar essa operação, deve-se consultar os metadados relacionados ao arquivo que deseja-se compartilhar para verificar se existem e quais as informações que já estão disponíveis nos provedores de nuvens públicas.
- **recoverPrivateData(**$Metadata$**):** Para recuperar um determinado arquivo, deve-se especificar o metadado ($Metadata$) do arquivo desejado. Assim como no método de

compartilhar, deve-se primeiramente tentar obter os metadados do arquivo que deseja-se recuperar para verificar quais as especificações do arquivo que deseja-se obter.

A implementação foi feita em módulos e serve como prova de conceito para validar as ideias aqui propostas, sendo assim, podem ser otimizadas para alcançar melhores resultados. Foram implementadas apenas as operações criptográficas para salvar e ler em disco. Portanto, não existe nenhum tipo de comunicação com servidores de nuvem para autenticação, envio ou recebimento de arquivos. Para a criptografia simétrica foi utilizado o algoritmo AES para cifrar os arquivos com um tamanho de chave de 128 bits. Para o resumo criptográfico foi utilizado o algoritmo SHA-1. NA cifragem das chaves simétricas foi utilizada a técnica de BF-IBE. Para assinar as partes cifradas foi utilizada a técnica de assinatura de Hess. Para os testes foram utilizados um total de quatro nodos, contando com um número de nodos redundantes igual a dois (este número está diretamente ligado ao processo de inicialização dos Distribuidores de Chaves Privadas (DCPs), do qual requerem um número mínimo de $3f + 1$, sendo $f + 1$ o quorum para recuperar a chave mestra), sendo f o número máximo de falhas toleradas no sistema.

Os protocolos foram implementados em C e C++. A implementação foi dividida em três partes: A inicialização dos DCP, o compartilhamento distribuído de documentos e os algoritmos de Leitura e Escrita. A inicialização dos DCPs foi implementada por Aniket em seu trabalho utilizando C++ e o protocolo modificado de JF-DKG. O trabalho de Aniket utiliza a biblioteca de emparelhamento bilinear PBC (*Pairing Based Cryptography*) [Lynn 2013]. As comunicações necessárias entre os clientes e servidores foram implementadas utilizando *sockets* e para a comunicação segura foi utilizada a biblioteca *OpenSSL*. Os algoritmos de Leitura e Escrita foram implementados em C/C++ utilizando as seguintes bibliotecas: *pbc library* [Lynn 2013] para realizar as operações de emparelhamento bilinear, a biblioteca *gfshare* para realizar as operações de segredo compartilhado de Shamir, a biblioteca *jerasure* para codificar e decodificar utilizando o *information-optimal erasure code* [Plank et al. 2008] e a biblioteca *OpenSSL* para realizar algumas das operações usuais de criptografia.

4. Algoritmos

O algoritmo 1 (Compartilhar Dado) autentica o usuário e solicita para o controle de acesso os metadados do documento (linha 5). O novo documento a ser armazenado terá a última versão encontrada (linha 6) mais um (linha 7). Uma chave simétrica é aleatoriamente gerada (linha 8) para cifrar o documento (linha 10). Um identificador ID para o documento é definido (linha 11) e uma chave pública baseada neste ID é criada (linha 12) utilizando a esquema de BF-IBE. A chave simétrica é então cifrada com a chave pública (linha 13) e então quebrada em pedaços utilizado o segredo compartilhado de Shamir (linha 14). O documento cifrado é então codificado utilizando-se o algoritmo de *information-optimal erasure code* (linha 15), reduzindo o tamanho dos dados que serão armazenados nos provedores de nuvem. A chave privada é então criada baseada no identificador ID (linha 17) de acordo com o procedimento da seção 2.2. Para cada parte das chaves e documentos cifrados, serão fornecidos os resumos criptográficos (linhas 19 e 20) e estes serão assinados (linhas 21 e 22) utilizando-se o esquema de assinatura Hess. Um bloco de dados é então criado, reunindo toda a informação necessária para armazenar o documento (linha 23). O bloco de dados é então enviado para os provedores de nuvem (linha 24) e a entrada deste armazenamento é enviado para o controle de acesso (linha 27).

Algoritmo 1 COMPARTILHARDADO($nomeArquivo, grupoArquivo, usuario, senha$)

```
1:  total ⟵ n
2:  redundante ⟵ m
3:  dado_ver ⟵ 0
4:  token ⟵ autenticar(usuario, senha)
5:  mt ⟵ buscarMetadados(nomeArquivo, grupoArquivo, usuario, token)
6:  dado_ver ⟵ max(mt[i].ver : 0 ≤ i ≤ n − 1)
7:  dado_ver_novo ⟵ dado_ver + 1
8:  ks ⟵ gerarChaveSim()
9:  dado ⟵ abrirArquivo(nomeArquivo)
10: e_dado ⟵ cifrar(dado, ks)
11: id ⟵ nomeArquivo + "/" + grupoArquivo + "/"+ dado_ver_novo
12: pubk_id ⟵ gerar_chave_pub(id)
13: e_ks ⟵ cifrar(ks, pubk_id)
14: enc_ks[0 .. n-1] ⟵ partir(e_ks, total − redundante, total)
15: enc_dado[0 .. n-1] ⟵ codificar(e_dado, total − redundante, redundante)
16: i ⟵ 0
17: privk_id ⟵ gerar_chave_priv(id)
18: for (0 ≤ i ≤ total − 1) do
19:     dado_hash ⟵ H(enc_data[i])
20:     ks_hash ⟵ H(enc_ks[i])
21:     dado_hash_assinado ⟵ assinar(dado_hash, privk_id)
22:     ks_hash_assinado ⟵ assinar(ks_hash, privk_id)
23:     blocoDeDado ⟵ (id, enc_ks[i], enc_dado[i], dado_hash_assinado, ks_hash_assinado)
24:     ack ⟵ mensagemDeEnviarDados(cloud_i, blocoDeDado, token)
25:     if (ack =′ ok′) then
26:         controleBlocoDeDado ⟵ (id, usuario, grupoArquivo)
27:         enviarMsgControleDeAcesso(cloud_i, controleBlocoDeDado)
28:     end if
29:     i ⟵ i + 1
30: end for
```

O algoritmo 2 (Recuperar Dado) primeiramente autentica o usuário e busca no controle de acesso pelos metadados do documento (linha 4). A última versão é então escolhida (linha 5) e o identificador ID é composto (linha 6). Um par de chaves é gerado a partir do ID utilizando o esquema BF-IBE (linhas 7 e 8) e a requisição pelos dados é iniciada (linha 11). Os blocos de dados são requeridos dos provedores de nuvem (linha 12), onde cada bloco de dado é verificado a partir das suas assinaturas e resumos criptográficos utilizando o esquema de assinatura Hess (linha 15). Após obter o número mínimo de blocos de dados (linha 23), as partes do documento cifrado são então decodificadas (linha 31), a chave simétrica é recomposta utilizando o segredo compartilhado de Shamir (linha 32), decifrada (linha 33) e finalmente, o documento original é decifrado (linha 34) e retornado para a aplicação.

5. Avaliação

5.1. Avaliação da Arquitetura

Baseado na pesquisa realizada, pode-se estabelecer os principais requisitos para elaborar uma arquitetura de middleware para garantir o compartilhamento sigiloso de documentos. Os requisitos necessários para o gerenciamento das chaves criptográficas e as soluções encontradas por este middleware proposto foram:

Custódia das Chaves: Apenas o usuário deverá ter posse da sua chave, evitando assim que as entidades distribuidoras de chaves possam ter posse das mesmas. Um dos principais problemas apontado pelo documento SP800-144 [Jansen and Grance 2011] é a vulnerabilidade de ataques internos e a falta de suporte legal em casos de intrusão devido a localização geográfica dos servidores. Para resolver este problema, este artigo propõe o uso de multi autoridades de CBI embarcados em módulos de segurança criptográficos (MSCs) utilizando o protocolo modificado de JF-DKG para

Algoritmo 2 RECUPERARDADO($nomeArquivo, grupoArquivo, usuario, senha$)

```
1: total ⟵ n
2: redundante ⟵ m
3: token ⟵ autenticar(usuario, senha)
4: mt ⟵ buscarMetadados(nomeArquivo, grupoArquivo, usuario, token)
5: dado_ver ⟵ max(mt[i].ver : 0 ≤ i ≤ n − 1)
6: id ⟵ nomeArquivo + "/" + grupoArquivo + "/" + dado_ver
7: privk_id ⟵ gerar_chave_privada(id)
8: pubk_id ⟵ gerar_chave_pub(id)
9: i ⟵ 0
10: ERRO ⟵ 0
11: while (i ≤ n − 1) do
12:     t_b ⟵ cloud_i.buscarBlocoDeDado(id, token)
13:     t_eks ⟵ t_b.retorna_enc_ks()
14:     t_edado ⟵ t_b.retorna_enc_data()
15:     rt ⟵ verifica(t_b.ks_hash_assinado_i, t_b.data_hash_assinado_i, t_eks, t_edado, pubk_id)
16:     if (rt = true) then
17:         enc_ks[i] ⟵ t_eks
18:         enc_dado[i] ⟵ t_edado
19:     else
20:         ERRO ⟵ ERRO + 1
21:     end if
22:     i ⟵ i + 1
23:     if (i > redundante − 1) then
24:         Break
25:     else
26:         if (ERRO > redundante − 1) then
27:             retorna ERRO
28:         end if
29:     end if
30: end while
31: e_dado ⟵ decodificar(enc_dado, total − redundante, redundante)
32: e_ks ⟵ combinar(enc_ks, total − dedundante, total)
33: ks ⟵ decifrar(e_ks, privk_id)
34: retorna. Decifrar(e_dado, ks)
```

gerar a chave mestra sem que nenhuma das autoridades tenha controle total da mesma. Utilizando este esquema, dois parâmetros são atribuídos: t e N, onde N é o número total de autoridades e t representa o número mínimo de partes que serão necessárias para recuperar a chave privada. Ao embarcar as autoridades em MSCs, consegue-se obter níveis altos de segurança física e lógica devido à natureza dos mesmos. Com isso, diminui-se a probabilidade de um ataque às autoridades e exposição das chaves privadas. Mas mesmo em um caso excepcional de sucesso de um ataque, um agente malicioso precisa descobrir um total de t partes para recompor o segredo, reduzindo assim as chances de sucesso de um ataque. Cada MSC terá posse de apenas uma parte do segredo mestre s_i e com isso, mesmo que obtenham o identificador que será utilizado, terão acesso apenas a uma parte $S_iH(ID)$ do segredo do usuário. Para a reconstrução total da chave, o atacante deverá possuir um número mínimo t. Sem este número mínimo, o atacante não conseguirá executar a equação $D_{id} = \sum_{P_i \in O} \lambda_i s_i H(ID)$ para obter a chave privada do usuário.

Revogação do Acesso: Após a retirada de um usuário de um grupo, o mesmo não deve mais ter acesso aos documentos cifrados e não poderá decifrar novos documentos que foram compartilhados. Para manter a confidencialidade da informação e evitar problemas com a revogação das chaves privadas, é recomendado utilizar parâmetros adicionais para identificar a chave pública da Criptografia Baseada em Identidade (CBI). Uma parte da solução é não vincular uma chave por usuário, mas vincular grupos de usuários com documentos, tendo assim chaves semânticas. Este trabalho propõe o uso de identificadores contendo regras de acesso concatenado com o nome do documento e a versão do mesmo. Estas regras são verificadas pelos DCPs por meio do controle de acesso que deve ser realizado de maneira distribuída e de uma forma confiável. O nome do documento vincula uma chave pública a um documento específico. A versão faz com

que a cada modificação do documento, seja gerado um novo par de chaves. Essas propriedades são garantidas pelo uso de um algoritmo de resumo criptográfico na geração dos identificadores e chaves, onde ID será a composição destas regras e a chave privada será definida por: $D_id = sH(ID)$. Portanto, um membro que faz parte de um grupo e obteve acesso a chave privada para decifrar um documento em uma versão X, não conseguirá obter uma chave consequente para decifrar um documento na versão $X + 1$ caso não faça mais parte deste grupo. Se um usuário já obteve a chave privada, este em algum momento teve acesso a um documento em uma versão específica. Desta forma, não existe a necessidade de recifrar o conteúdo do documento que já foi exposto. Caso um usuário não tenha ainda obtido a chave privada e já não pertence mais a um determinado grupo, este não mais terá acesso às partes das chaves e documentos cifrados, pois o controle de acesso não permitirá mais o acesso devido ao não cumprimento das regras pré-estabelecidas para aquele documento. Para mais detalhes a respeito do controle de acesso, o trabalho de dissertação de Rick [de Souza 2014] pode ser consultado.

Tolerância a Falhas: O sistema deve fornecer dois níveis de tolerância a falha. Tanto para o gerenciamento de chaves quanto para o armazenamento de documentos cifrados. Caso um servidor fique indisponível, deve-se prover mecanismos de contingência dos servidores para manter a solução em funcionamento. Utilizando DCPs distribuídos, segredo compartilhado e *erasure code*, consegue-se garantir tolerância a falhas. Este trabalho propõe protocolos baseados no quorum secreto compartilhado e verificável, aumentando assim o rigor na verificação das partes distribuídas. A tolerância a falhas bizantinas e a disponibilidade são garantidas com a propriedade de que em um total de $3f + 1$ DCPs, apenas f podem falhar e consequentemente, são necessários $2f + 1$ servidores para garantir o correto funcionamento do sistema, garantindo assim a disponibilidade e confiança. Desta forma, mesmo que f DCPs falhem, ao obter $f + 1$ respostas do restante dos servidores, pode-se reconstruir a chave privada do usuário utilizando-se do método de extração de chaves privadas mencionado na sub-sessão 2.2.

5.2. Análise de Performance

Com a implementação pode-se avaliar o desempenho dos algoritmos Compartilhar Dado e Recuperar Dado, mencionados na seção 4. Foram feitas sequências de execuções utilizando diferentes tamanhos de arquivos. Os testes partiram de arquivos com tamanho de 1 Kbyte até 524288 Kbytes (512 MBytes). Os testes foram executados em um computador com as seguintes características: Processador Intel i3, 4GB RAM com um sistema operacional Linux Ubuntu. Os testes foram executados durante uma semana e cada algoritmo avaliado conforme o desvio padrão. Entre 1 Kbyte e 16384 Kbytes o tempo foi instável, superando 5% de desvio padrão; isto se deve ao pequeno tamanho dos arquivos. Para arquivos de tamanho entre 1 Kbyte e 16384 Kbytes os tempos variaram até um valor máximo de 300 milissegundos. A partir de 16384 Kbytes, o tempo começou a crescer linearmente dobrando conforme o tamanho do arquivo. A figura 3 ilustra o comportamento conforme o aumento do tamanho dos arquivos entre 1 Kbyte e 8 Mbytes e a figura 4 ilustra o desempenho dos arquivos entre 16 Mbytes e 512 Mbytes.

6. Conclusão

Com a arquitetura proposta atingiu-se os objetivos propostos inicialmente de criar um middleware cujas funcionalidades integrassem os serviços de gerenciamento seguro de

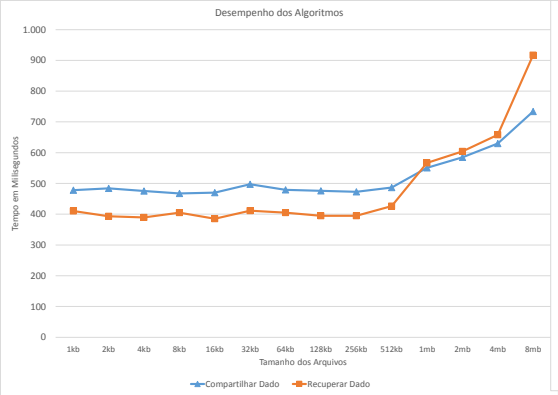


Figura 3. Desempenho para Compartilhar e Recuperar Dado com arquivos entre os tamanhos de 1Kbyte e 8Mbytes.

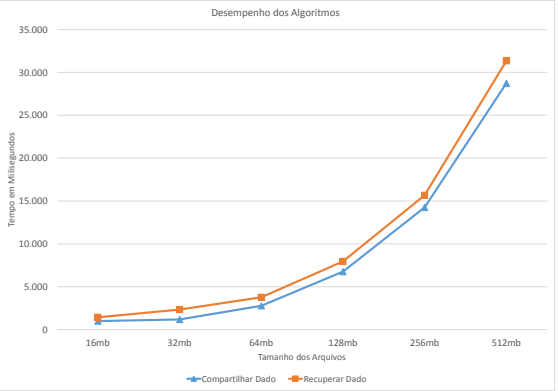


Figura 4. Desempenho para Recuperar e Compartilhar dado para arquivos entre os tamanhos de 16Mbytes e 512Mbytes.

chaves criptográficas juntamente com o compartilhamento de documentos em diferentes provedores de nuvem pública. Para o gerenciamento das chaves utilizou-se módulos de segurança criptográficos distribuídos, fazendo com que os mesmos possuam segurança física e lógica, garantindo a integridade das chaves mestras utilizadas para gerar as chaves privadas dos usuários. Para a segurança e tolerância a falhas no armazenamento das partes dos documentos, a arquitetura utiliza diferentes provedores de nuvem pública para armazenamento. Dessa forma, ao utilizar uma nuvem híbrida, consegue-se garantir o nível de segurança necessário para o compartilhamento seguro de documentos sigilosos.

A garantia de segurança em caso de saída e entrada de membros dos grupos foi garantida por meio do gerenciamento de grupos juntamente com o nome utilizado na geração das chaves criptográficas baseadas em identidade. Ao utilizar um identificador que é composto por nome do documento, grupo custodiante e versão, consegue-se alcançar um nível de unicidade suficiente para a garantia de segurança nos casos de saída e entrada de um novo membro. Dessa forma, o controle de acesso juntamente com as propriedades de segurança da criptografia baseada em identidade garantem a segurança na revogação das chaves privadas.

O uso de um conjunto de geradores de chaves privadas distribuídos garante segurança contra das chaves privadas e também garante tolerância a falhas. Dessa forma, ao estabelecer um procedimento de inicialização dos geradores de chaves privadas (n, t), consegue-se estabelecer um número mínimo t de partes necessárias para recompor as chaves privadas, tal que $n \geq t$, de tal forma que apenas o usuário que se autenticar de maneira correta em t gerenciadores de chaves possa recuperar sua chave. Isso também garante a tolerância a falhas, devido ao fato de que caso o sistema feito seja $n = 4$ e $t = 3$, mesmo com um servidor comprometido, pode-se conseguir recompor a chave privada utilizando os outros três servidores.

Referências

Bessani, A., Correia, M., Quaresma, B., André, F., and Sousa, P. (2011). Depsky: dependable and secure storage in a cloud-of-clouds. In *Proceedings of the sixth conference*

on Computer systems, pages 31–46. ACM.

Boneh, D. and Franklin, M. (2001). Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001*, pages 213–229. Springer.

de Souza, R. L. (2014). Um Middleware para Compartilhamento de Documentos Sigilosos em Nuvens Computacionais. Master's thesis, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina.

Hess, F. (2003). Efficient identity based signature schemes based on pairings. In *Selected Areas in Cryptography*, pages 310–324. Springer.

Itani, W., Kayssi, A., and Chehab, A. (2009). Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In *International Conference on Dependable, Autonomic and Secure Computing*, pages 711–716. IEEE.

Jansen, W. and Grance, T. (2011). Guidelines on security and privacy in public cloud computing. *NIST special publication*, pages 800–144.

Jung, T., Li, X.-Y., Wan, Z., and Wan, M. (2013). Privacy preserving cloud data access with multi-authorities. In *IEEE INFOCOM*.

Kate, A., Huang, Y., and Goldberg, I. (2012). Distributed key generation in the wild. *IACR Cryptology ePrint Archive*, 2012:377.

Lynn, B. (Novembro, 2013). The pairing-based cryptography (pbc) library. *Available on http://crypto. stanford. edu/pbc*.

Padilha, R. and Pedone, F. (2011). Belisarius: Bft storage with confidentiality. In *Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*, pages 9–16. IEEE.

Pearson, S., Shen, Y., and Mowbray, M. (2009). A privacy manager for cloud computing. In *Cloud Computing*, pages 90–106. Springer.

Plank, J. S., Simmerman, S., and Schuman, C. D. (2008). Jerasure: A library in c/c++ facilitating erasure coding for storage applications-version 1.2. *University of Tennessee, Tech. Rep. CS-08-627*, 23.

Ruj, S., Nayak, A., and Stojmenovic, I. (2011). Dacc: Distributed access control in clouds. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 91–98. IEEE.

Singh, Y., Kandah, F., and Zhang, W. (2011). A secured cost-effective multi-cloud storage in cloud computing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 619–624. IEEE.

Yang, K., Jia, X., and Ren, K. (2012a). Dac-macs: Effective data access control for multi-authority cloud storage systems. *IACR Cryptology ePrint Archive*, 2012:419.

Yang, K., Liu, Z., Cao, Z., Jia, X., Wong, D. S., and Ren, K. (2012b). Taac: Temporal attribute-based access control for multi-authority cloud storage systems. *IACR Cryptology ePrint Archive*, 2012:651.

Zhou, L., Varadharajan, V., and Hitchens, M. (2011). Enforcing role-based access control for secure data storage in the cloud. *The Computer Journal*, 54(10):1675–1687.

Análise de cerimônias no sistema de votação Helios

Taciane Martimiano[1], **Jean Everson Martina**[1], **M. Maina Olembo**[2]

[1]Universidade Federal de Santa Catarina

[2]Technische Universität Darmstadt

{taciane.m,everson}@inf.ufsc.br, maina.olembo@cased.de

Abstract. *Helios is an online voting system, which allows its voters to verify whether their votes were correctly computed by the booth and stored for final tally. Usability improvements were proposed by Neumann, they are the use of a) several independent verifying institutes and b) smartphone app developed individually by these institutes, for verifying processes and correct storage of the submitted votes on the bulletin board. In this work, these improvements are analyzed as security ceremonies, based on the adaptive threat model proposed by Carlos et al.*
Key words: Ceremony analysis, threat models, usability, Helios

Resumo. *Helios é um sistema de votação online, que permite aos eleitores verificarem se seu voto foi corretamente computado pela cabine de votação e armazenado para contabilização final dos votos. Melhorias visando a usabilidade foram propostas por Neumann, são elas o uso de a) diversos independentes institutos de verificação e b) aplicativo para smartphone desenvolvido individualmente por tais institutos, para processos de verificação e armazenamento correto do voto no quadro de avisos. No presente trabalho, tais melhorias são analisadas como cerimônias de segurança, com base no modelo de ameaça adaptativo proposto por Carlos et al.*
Palavras-chave: Análise de cerimônias, modelos de ameaça, usabilidade, Helios

1. Introdução

Visando a confiança dos eleitores, sistemas criptográficos de votação online que oferecem verificabilidade, enquanto mantém o sigilo do voto, tem sido propostos e continuam sendo aprimorados. Nesse contexto, Helios[Adida 2008][Adida et al. 2009], um sistema de votação baseado na Internet, verificável e de código aberto, tem sido usado principalmente no meio acadêmico.

Levando a usabilidade do sistema em consideração, melhorias foram sugeridas como tentativa de estimular o uso correto e prático do sistema. Para tanto, o eleitor pode usar as páginas web dos institutos confiáveis participantes ou baixar e instalar um aplicativo em seu smartphone[Neumann et al. 2014]. Uma análise da segurança computacional de tais propostas é desenvolvida nesse trabalho, com foco em sigilo e integridade (propriedades importantes para a votação verificável). Para isso, as propostas foram modeladas como cerimônias de segurança, empregando o framework proposto por Carlos et al[Carlos et al. 2013], o qual é baseado no conjunto de capacidades do modelo de ameaça de Dolev-Yao[Dolev and Yao 1983].

O trabalho está estruturado da seguinte forma: o capítulo 2 aborda definições importantes para a compreensão das análises[1] apresentadas posteriormente. Nesse capítulo encontra-se a definição do conceito de cerimônias e seus meios de comunicação, modelos de ameaça, assim como suposições consideradas para realização das análises. O capítulo 3 apresenta a proposta ao sistema Helios versão web com uso de institutos de verificação. A seguir, apresenta a análise dessa proposta e os resultados alcançados. O capítulo 4 apresenta a proposta ao sistema Helios fazendo uso dos aplicativos para smartphone da eleição. A seguir, apresenta a análise dessa proposta, os resultados alcançados e vantagens em se empregar cerimônias. Por fim, o capítulo 5 traz as conclusões e trabalhos futuros.

2. Conceitualização

Nesta seção está definido o conceito de cerimônias e apresentado o modelo de ameaça adaptativo utilizado para análise das cerimônias.

Análise de cerimônias e modelo de ameaça adaptativo

Cerimônias e protocolos de segurança podem ser definidos como uma sequência de interações entre entidades com o intuito de atingir um certo objetivo, como por exemplo autenticação de entidades, distribuição de chaves, etc. A análise de cerimônias estende a análise de protocolos devido à inclusão de nodos humanos ao sistema[Ellison 2007]. Tal inclusão implica em um aumento de complexidade da análise, contudo proporciona resultados mais precisos e completos, sendo possível inclusive detectar falhas de segurança previamente não detectáveis[Carlos et al. 2013].

Nos protocolos, as ações humanas são meramente modeladas como suposições. Quando o protocolo é então implementado, essas suposições resultam em interações de usuário não realísticas (não compatíveis com situações cotidianas do mundo real). Em uma cerimônia de segurança, tem-se o fator humano projetado como parte integrante do sistema. Para isso, as cerimônias apresentam dois canais adicionais ao tradicional canal dispositivo-dispositivo (DD) (proveniente da estrutura dos protocolos). Esses canais adicionais são o canal humano-dispositivo (HD) e o canal humano-humano (HH), empregados para relacionar o 'nodo' humano aos demais nodos do sistema[Carlos et al. 2013]. Contudo, cerimônias de segurança ainda precisam de algumas suposições, tais como conhecimento inicial dos agentes humanos. Essas suposições tendem a ser mais detalhadamente descritas e realísticas em comparação às suposições dos protocolos.

Um dos desafios relativos à análise de cerimônias é a definição do perfil do atacante. O modelo mais conhecido e adotado é o modelo Dolev-Yao. Dolev e Yao[Dolev and Yao 1983] formalizaram o modelo de atacante introduzido por Needham and Schroeder[Needham and Schroeder 1978], onde o atacante tem total controle da rede, sendo capaz de copiar, replicar, alterar e criar mensagens. Ao atacante só não é permitido fazer criptoanálise. Porém, cada cerimônia deve ser analisada individualmente para obtenção do modelo mais adequado e realístico, dados os cenários que envolvem tal cerimônia. Caso as capacidades do atacante sejam exageradas, provavelmente a cerimônia será extremamente complexa e inutilizável. Entretanto, se as capacidades do atacante forem subestimadas, a cerimônia será falha. Para lidar com essa questão, o modelo de ameaça adaptativo proposto por Carlos et al é utilizado na análise das cerimônias apresentadas. Segundo esse modelo adaptativo, a cerimônia pode começar com um modelo de

[1] As análises contidas neste trabalho são de caráter semiformal, inclusive devido à limitação de espaço.

ameaça Dolev-Yao, a partir do qual removem-se capacidades (do conjunto de todas as capacidades que um atacante Dolev-Yao possui) a fim de tornar o atacante mais realístico. Entender o correto modelo de ameaça (ao qual o usuário estará sujeito ao interagir em uma cerimônia) evita sobrecarregar tal usuário com situações hipotéticas e garante que as propriedades de segurança estritamente necessárias sejam empregadas. Para cada canal será definido um modelo de ameaça, de acordo com as capacidades realísticas definidas sobre o conjunto total de capacidades de um atacante Dolev-Yao definidas em Carlos et al[Carlos et al. 2013].

O processo de análise começa com o estabelecimento dos canais presentes na cerimônia. Isso envolve listar os nodos humanos e dispositivos envolvidos, identificar quais desses nodos trocam mensagens entre si, e que tipo de canal essa comunicação caracteriza (isto é, HH, HD ou DD). Assim, é possível analisar o impacto das capacidades de um atacante em cada um dos canais. O atacante objetiva aprender o conhecimento trocado entre nodos. O modelo Dolev-Yao (DY) define habilidades que permitem ao atacante alcançar tal objetivo. Portanto, observa-se em cada mensagem quais abordagens o atacante pode usar para barrar ou modificar mensagens, criar e enviar mensagens de seu próprio conhecimento, etc. de forma a se obter um modelo de ameaça realístico que engloba os perfis dos atacantes associados a cada canal. Por exemplo, se o atacante tem acesso a uma dada chave criptográfica e intercepta mensagens cifradas com essa chave, ele será capaz de decifrar e aprender os conteúdos dessas mensagens, comprometendo a segurança das informações compartilhadas por tal canal[2]. É interessante ressaltar que seguindo o modelo de ameaça adaptativo de Carlos et al, temos um modelo de ameaça realístico e específico para cada cerimônia, dados seus participantes, canais e circunstâncias às quais estará sujeita.

Suposições

O presente trabalho utiliza algumas suposições relacionadas ao Helios e às entidades e análises das cerimônias, citadas a seguir: Considera-se que as entidades presentes nas cerimônias são confiáveis no que diz respeito à integridade do processo sendo executado (suposição já presente no Helios original[Adida 2008]). O atacante está presente nos canais de comunicação, sendo essa uma suposição típica do modelo Dolev-Yao. A cabine de votação do Helios é confiável e, assim, o eleitor tem motivação em usar o sistema para votar e verificar seu voto. Os institutos participantes são confiáveis, uma vez que qualquer comportamento malicioso pode conduzir à perda de reputação. O eleitor é um 'nodo' honesto na cerimônia, pois um eleitor desonesto pode facilmente corromper a conclusão correta da cerimônia em questão [3]. Além disso, o eleitor confia na cabine de votação quanto ao sigilo das informações. A cerimônia é considerada como tendo um único ponto de início e um único ponto de saída. O eleitor deve seguir todos os passos previstos na cerimônia que ele está executando.

3. Aplicação web vinculada a institutos de verificação

Esta seção aborda brevemente os processos que os eleitores desempenhariam utilizando verificação provida pelos institutos de confiança. Através do modelo adaptativo, é

[2]As capacidades do atacante DY aqui referidas encontram-se em [Carlos et al. 2013]

[3]Neste trabalho, não considera-se coerção. Assim, não foram incluídos casos em que o atacante é o próprio eleitor.

possível enfatizar como a presença humana limita as ações de possíveis atacantes do sistema.

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$
10.	V	$\xrightarrow[HD_1]{}$	B	:	"*Check encrypted vote*"
11.	B	$\xrightarrow[HD_1]{}$	V	:	Verifying Options Page
12.	V	$\xrightarrow[HD_1]{}$	B	:	I
13.	B	$\xrightarrow[DD_1]{}$	I	:	Vote + R
14.	I	$\xrightarrow[HD_2]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$ + Vote

Figura 1. Voto de teste

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$
10.	V	$\xrightarrow[HD_1]{}$	B	:	"*Log in to cast vote*"
11.	B	$\xrightarrow[HD_1]{}$	V	:	Login Page
12.	V	$\xrightarrow[HD_1]{}$	B	:	VoterID + Password
13.	B	$\xrightarrow[DD_3]{}$	BB	:	$\{Vote\}_{Epk}$
14.	B	$\xrightarrow[HD_1]{}$	V	:	Vote is submitted
15.	V	$\xrightarrow[HD_1]{}$	B	:	I
16.	B	$\xrightarrow[DD_1]{}$	I	:	$H\{\{Vote + R\}_{Epk}\}$
17.	I	$\xrightarrow[HD_2]{}$	V	:	$H\{\{Vote + R\}_{Epk}\}$ + VoterID

Figura 2. Voto final

Nessa proposta[4], o processo de votação é similar ao processo do Helios original. As diferenças surgem nos processos de verificação do voto. Para poder verificar se o voto está corretamente cifrado, o eleitor primeiramente precisa registrar o hash apresentado pela cabine de votação. O eleitor então expressa sua intenção de verificar o voto e seleciona um instituto no qual confia. A cabine de votação transmite as informações necessárias para verificação para o instituto. O instituto, por sua vez, irá computar o hash a partir das informações recebidas da cabine e apresentará o resultado de suas computações ao eleitor, juntamente com o voto recebido. O eleitor agora precisa checar e confirmar se os dois hashes são iguais e se o voto apresentado é de fato correspondente ao candidato escolhido.

Para o eleitor conferir se o seu voto final foi corretamente armazenado no quadro de avisos, ele registra o hash apresentado. O eleitor, então, faz log in para submeter seu voto. Após autenticação bem sucedida, a cabine de votação submete o voto ao quadro de avisos. O eleitor seleciona um instituto dos vários disponíveis, após ter submetido seu voto. Uma nova página web abre, onde o eleitor entra com o hash registrado anteriormente e confere o resultado apresentado pelo instituto. O instituto também precisa apresentar o ID do eleitor, para evitar problemas de colisão de hashes[Kusters et al. 2012]. Assim, mesmo que para dois eleitores seja apresentado o mesmo hash, pelo ID (que é único para cada eleitor) tal problema pode ser identificado.

3.1. Análise

Para o conhecido modelo de ameaça Dolev-Yao (DY), todos os canais de comunicação estão sob um atacante DY. Já no modelo de ameaça adaptativo, considera-se que apenas

[4]As cerimônias que ilustram a proposta estão nas figuras 1 e 2. As mensagens estão em inglês para fazer jus ao sistema real ao qual se referem. As entidades presentes são o eleitor (representado pela letra V, *Voter* em inglês), cabine de votação (letra B, *Booth* em inglês), instituto (letra I, *Institute* em inglês) e quadro de avisos (letras BB, *Bulletin Board* em inglês). As letras abaixo das setas representam o canal pelo qual a mensagem (apresentada ao lado direito de cada imagem) é transmitida.

o canal dispositivo-dispositivo (DD) está sob um atacante Dolev-Yao (DY), enquanto que os canais humano-humano (HH)[5] e humano-dispositivo (HD) estão sob um atacante DY-E. DY-E significa que tal atacante possui todas as capacidades de um atacante DY, exceto a capacidade Escuta (*Eavesdrop*). Essa capacidade é excluída, pois consideram-se ambientes controlados onde o eleitor não precisa checar ao seu redor e assegurar-se de que não há alguém espionando-o. A respeito do canal HD, assume-se que existe um ser humano (e não uma máquina fingindo ser um humano) lidando com um dispositivo (por exemplo, olhando para a tela e digitando algo no teclado). Assim, um atacante DY-E não é capaz de comprometer o sigilo das mensagens enviadas por canais HD uma vez que o eleitor está no domínio do dispositivo, limitando as ações do atacante. Por exemplo, mesmo que o atacante possua as capacidades de Fabricar (*Fabricate*) e Criptografia (*Crypto*), ele só pode aplicar tais capacidades em mensagens e conhecimento que ele já possua. Portanto, não há ameaças nas cerimônias estudadas dado que o atacante não é capaz de aprender nenhuma informação no que diz respeito às ações do eleitor.

Considerando o modelo de ameaça DY, o atacante tem total controle de todos os canais de comunicação e é capaz de manipular o eleitor durante todo o processo. Em tais cenários, o atacante intercepta todas as mensagens trocadas entre os pares de nodos do sistema, e envia mensagens de seu próprio conhecimento no lugar das originais. Simultaneamente, para o eleitor são apresentados dados corretos, onde o atacante se faz passar pelas entidades legítimas. Portanto, o eleitor é levado a acreditar que seu voto foi cifrado, submetido e armazenado apropriadamente, quando isso não é verdade. Contudo, essa situação é altamente improvável de acontecer nas cerimônias apresentadas, pois o canal HD limita as ações do atacante.

Já um cenário realístico e factível é o atacante interceptar mensagens apenas no canal DD. Nesse caso, o instituto recebe informações alteradas, calculando um hash diferente do esperado pelo eleitor. O eleitor passa a não confiar mais no instituto. Esse resultado ressalta a necessidade de se ter vários institutos disponíveis, provendo serviços de verificação para os eleitores. Portanto, o eleitor tem total liberdade de verificar usando vários outros institutos. Se as tentativas seguintes também falharem, então o eleitor pode contatar a comissão eleitoral.

Analisando as cerimônias apresentadas acima, constata-se que a mensagem 13 da cerimônia Voto de teste (figura 1) apresenta Voto + R sendo transmitido como texto plano (sem criptografia) através de um canal DD. O mesmo não acontece com a cerimônia Voto final (figura 2), onde tal mensagem contém informações cifradas com a chave pública da eleição (E_{pk}). Logo, o sigilo não está presente para a cerimônia de voto de teste, estando presente apenas na cerimônia para voto final. Tal conclusão se deve ao fato de que o atacante não aprende o voto ou a informação randômica a partir do hash.

[5]Neste trabalho, nenhuma das cerimônias abordadas utiliza canal humano-humano (HH), assim apenas o canal humano-dispositivo(HD) será abordado.

4. Aplicativo para smartphone

Nessa proposta[6], o eleitor verifica o voto usando um dispositivo diferente do usado para votar, assim não há mais a necessidade de que o eleitor confie no dispositivo que utiliza para votar. Tal dispositivo para verificação é o próprio smartphone do eleitor, que já está em seu domínio e com o qual ele já está acostumado. Esta seção aborda brevemente os processos que os eleitores desempenhariam utilizando verificação provida pelo aplicativo da eleição no smartphone.

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	Check-code + $QRCode^*$
10.	V	$\xrightarrow[HD_3]{}$	A	:	"*Open/Scan*"
11.	B	$\xrightarrow[DD_2]{}$	A	:	QRCode
12.	V	$\xrightarrow[HD_1]{}$	B	:	"*Check encrypted vote with app*"
13.	B	$\xrightarrow[HD_1]{}$	V	:	QRCode Page
14.	V	$\xrightarrow[HD_3]{}$	A	:	"*Check that vote is correctly encrypted*"
15.	B	$\xrightarrow[DD_2]{}$	A	:	$QRCode^{**}$
16.	A	$\xrightarrow[HD_3]{}$	V	:	"*Check-codes match. Is this your vote?*" + Vote
17.	V	$\xrightarrow[HD_3]{}$	A	:	Confirm (Yes)
18.	V	$\xrightarrow[HD_3]{}$	A	:	"*Close*"

*Check-code is the hash $H\{\{Vote + R\}_{Epk}\}$. In messages 9 and 11, the QRCode has the check-code contents.

**In messages 13 and 15, the QRCode has $(Vote + R + E_{pk})$ information.

Figura 3. Voto de teste

1.	V	$\xrightarrow[HD_1]{}$	B	:	URL
2.	B	$\xrightarrow[HD_1]{}$	V	:	Instructions Page
3.	V	$\xrightarrow[HD_1]{}$	B	:	"*Proceed to ballot*"
4.	B	$\xrightarrow[HD_1]{}$	V	:	Ballot Page
5.	V	$\xrightarrow[HD_1]{}$	B	:	Vote
6.	V	$\xrightarrow[HD_1]{}$	B	:	"*Review vote*"
7.	B	$\xrightarrow[HD_1]{}$	V	:	Review Page
8.	V	$\xrightarrow[HD_1]{}$	B	:	"*Encrypt vote*"
9.	B	$\xrightarrow[HD_1]{}$	V	:	Check-code + $QRCode^*$
10.	V	$\xrightarrow[HD_3]{}$	A	:	"*Open/Scan*"
11.	B	$\xrightarrow[DD_2]{}$	A	:	QRCode
12.	V	$\xrightarrow[HD_1]{}$	B	:	"*Log in to cast vote*"
13.	B	$\xrightarrow[HD_1]{}$	V	:	Login Page
14.	V	$\xrightarrow[HD_1]{}$	B	:	VoterID + Password
15.	B	$\xrightarrow[DD_3]{}$	BB	:	$\{Vote\}_{Epk}$
16.	B	$\xrightarrow[HD_1]{}$	V	:	Vote is submitted
17.	V	$\xrightarrow[HD_3]{}$	A	:	"*Check that vote is on bulletin board*"
18.	A	$\xrightarrow[DD_4]{}$	BB	:	$H\{\{Vote + R\}_{Epk}\}$
19.	BB	$\xrightarrow[DD_4]{}$	A	:	Confirmation + VoterID

*Check-code is the hash $H\{\{Vote + R\}_{Epk}\}$. In messages 9 and 11, the QRCode has the check-code contents.

Figura 4. Voto final

A cabine de votação apresenta um QR code contendo o hash do voto, juntamente com o próprio hash. O eleitor utiliza seu smartphone para escanear o QR code que contém esse hash. Tal hash será armazenado pelo aplicativo para uso posterior no processo de verificação do voto. O eleitor expressa sua intenção de verificar o voto para a cabine de votação. A seguir, ele escaneia um segundo QR code e o aplicativo computa o hash e o compara com o armazenado anteriormente. Em um caso de comparação bem sucedida, o aplicativo informa que os hashes são iguais e pede ao eleitor para confirmar que o voto apresentado na tela é o voto correto.

Para verificar que o voto foi corretamente armazenado para contagem final no quadro de avisos (cerimônia apresentada na figura 4), o eleitor escaneia o primeiro QR code que contém o hash. O eleitor faz log in e a cabine de votação submete seu voto após autenticação bem sucedida. O eleitor utiliza o aplicativo para checar o quadro de avisos procurando pelo hash do seu voto. O aplicativo realiza essa checagem consultando o quadro de avisos pelo valor do hash. Em caso bem sucedido, o aplicativo mostra uma mensagem para o eleitor afirmando que o hash foi armazenado no quadro de avisos. Para

[6]As cerimônias que ilustram a proposta estão nas figuras 3 e 4. As mensagens estão em inglês para fazer jus ao sistema real ao qual se referem. Além das entidades que já apareceram nas cerimônias anteriores, tem-se a presença da entidade aplicativo (representado pela letra A, *App* em inglês).

prevenir problemas de colisão [Kusters et al. 2012], o aplicativo também retorna o ID do eleitor. Em caso mal sucedido, o aplicativo informa o eleitor de que o hash não foi encontrado no quadro de avisos. O eleitor pode usar outros aplicativos para verificação. Em caso de múltiplos hashes falharem, o eleitor pode entrar em contato com a comissão eleitoral.

4.1. Análise

Apesar de os canais DD geralmente estarem sob um atacante DY, isso não é realístico para o canal DD_2 (por exemplo, na mensagem 11 da figura 3). Esse canal é um 'canal visual' uma vez que não há bluetooth ou conexão de qualquer modo entre os dispositivos envolvidos. Nesse específico cenário, o eleitor usa seu smartphone para escanear o QR code apresentado pelo computador. Considera-se que ambos os dispositivos estão no domínio do eleitor e não sob controle do atacante. Para o canal DD_2, tem-se situações similares aos canais HD (descritas na seção 3). Por exemplo, o atacante não pode bloquear os conteúdos passando por esse canal pois isso implicaria o atacante bloquear a tela do computador do eleitor e seu smartphone. Situações similares acontecem se o atacante tenta aplicar qualquer outra de suas capacidades. Assim, para um ataque ser bem sucedido, o atacante precisa estar no domínio dos dispositivos do eleitor. Tal cenário só seria factível se o eleitor deixasse os dispositivos abandonados no meio do processo de votação. Portanto, considera-se que o canal DD_2 também é DY-E, assim como os canais HD. Qualquer combinação enfraquecida de capacidades do atacante DY (qualquer combinação de capacidades, não envolvendo Escuta) continuará não sendo efetiva em tais canais. Isso se deve ao fato de que Escuta é a única capacidade que pode comprometer o sigilo do voto do eleitor.

No que diz respeito ao modelo de ameaça DY, o atacante pode manipular o eleitor através da manipulação das informações apresentadas a ele. Tal situação pode ser considerada realística para a cerimônia do voto final usando aplicativo (figura 4). Contudo, é altamente improvável de acontecer devido ao fato dos canais HD estarem seguros sob a suposição de que o ambiente é controlado. Adicionalmente, foi demonstrado ser irrealista para a cerimônia de voto de teste usando aplicativo (figura 3). Tal cerimônia é mais segura por possuir o canal visual, o qual limita as ações do atacante, pois apresenta o mesmo comportamento que os canais HD. Uma contribuição muito importante da proposta usando o aplicativo constitui-se de que ambos os votos de teste e final são secretos, quando comparados com a proposta que faz uso dos institutos (onde o voto de teste é enviado sem uso de criptografia por um canal DY). Tal contribuição significa que essa cerimônia possui a propriedade do sigilo e, como as mensagens não são interrompidas e não são modificadas, conclui-se que tal cerimônia também garante integridade.

5. Conclusão

Neste trabalho foi analisada a segurança das propostas feitas para o sistema de votação Helios. Para tal análise foi utilizado o framework proposto por Carlos et al[Carlos et al. 2013], aplicando o modelo adaptativo de ameaça. Para esse fim, os processos de votação e verificação do voto foram abordados como cerimônias, integrando a interação humana na análise. O modelo de ameça Dolev-Yao foi usado para comparação com o modelo adaptativo, onde foi possível ressaltar os ganhos em se empregar um modelo que reflita as necessidades de segurança para cada específico cenário sem sobrecarregar o usuário.

Na primeira proposta para verificação, usando institutos confiáveis, os resultados mostram a possibilidade de violações de sigilo quando o eleitor verifica se seu voto está corretamente cifrado, e violações de integridade quando ele verifica se seu voto foi corretamente submetido no quadro de avisos. As violações de integridade tomam forma de 'ataques de reputação', resultando na perda de confiança no instituto por parte do eleitor (ao receber informações incorretas). Para as cerimônias envolvendo o aplicativo, o sigilo é mantido devido à presença de um canal visual e ao fato da informação ser enviada cifrada (e não em forma de texto plano). Os resultados também mostram que nenhum ataque significativo pode ocorrer quando o eleitor verifica se seu voto está cifrado de forma correta. Violações de integridade acontecem através dos 'ataques de reputação', cuja estratégia de mitigação é a existência de diversos institutos, ou aplicativos mantidos por esses, disponíveis para o eleitor. Através dessa solução, a propriedade da integridade pode ser mantida em ambas as propostas abordadas. No caso do processo de verificação falhar em algum dos casos, o eleitor pode verificar fazendo uso de outras fontes.

Os resultados desse trabalho ressaltaram várias melhorias que podem ser feitas ao protocolo de votação do Helios. Uma futura proposta envolve o eleitor entrar com uma informação única conhecida apenas por ele. Verificar a presença dessa informação em um estágio posterior do sistema garantiria ao eleitor a integridade do voto submetido. Propostas serão desenvolvidas com o objetivo de equilibrar os aspectos de segurança e as expectativas e habilidades dos nodos humanos na cerimônia.

Referências

Adida, B. (2008). Helios: Web-based Open-Audit Voting. In *Proceedings of the 17th Symposium on Security*, pages 335 – 348. Usenix Association.

Adida, B., De Marneffe, O., Pereira, O., and Quisquater, J.-J. (2009). Electing A University President using Open-Audit Voting: Analysis of Real-World Use of Helios. In *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, pages 10–10. Usenix Association.

Carlos, M. C., Martina, J., Price, G., and Custodio, R. F. (2013). An Updated Threat Model for Security Ceremonies. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 1836–1843, New York, NY, USA. ACM.

Dolev, D. and Yao, A. C. (1983). On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29(2):198–208.

Ellison, C. (2007). Ceremony Design and Analysis. Cryptology ePrint Archive, Report 2007/399. `http://eprint.iacr.org/`.

Kusters, R., Truderung, T., and Vogt, A. (2012). Clash Attacks on the Verifiability of E-Voting Systems. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 395–409.

Needham, R. M. and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999.

Neumann, S., Olembo, M. M., Renaud, K., and Volkamer, M. (2014). Helios verification: To alleviate, or to nominate: Is that the question, or shall we have both? In *3rd International Conference on Electronic Government and the Information Systems Perspective*. Springer. To appear.

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

Trilha principal — Artigos Curtos

Simo: Security Incident Management Ontology

Pâmela Carvalho da Silva[1], Leonardo Lemes Fagundes[1]

Universidade do Vale do Rio dos Sinos (UNISINOS)
CEP 93.022-000 – São Leopoldo – RS - Brasil

Resumo. *Os incidentes de segurança da informação apresentam características comuns, bem como variações quanto aos ataques, sua complexidade e sofisticação. Isto requer profissionais capacitados para desempenhar ações e atividades relacionadas a identificação, tratamento e prevenção de incidentes. O trabalho proposto apresenta uma abordagem para auxiliar e apoiar a capacitação de profissionais com o uso de uma ontologia de domínio para a gestão de incidentes de segurança da informação baseada na norma ISO/IEC 27035:2011.*

Abstract. *The information security incidents present common features, but also variations in the attacks, its complexity and sophistication. That requires trained professionals to perform actions and activities related to identification, treatment and prevention of incidents. The proposed project presents an approach to assist and support the training of professionals using a domain ontology for incident management of information security, based on ISO / IEC 27035: 2011.*

1. Introdução

O cenário de ataques expande-se em quantidade, diversidade, complexidade, sofisticação e subversividade (Mccarthy, 2014). Dentre as causas, destacam-se a constante e crescente dependência de sistemas e tecnologias da informação; a popularização de novas tecnologias; e a motivação e ousadia dos atacantes.

Este cenário, caracterizado pelo alto número de incidentes e pela sofisticação dos novos ataques, exige profissionais capacitados. A sofisticação das ameaças requer a estruturação de habilidades referentes a prevenção, identificação e tratamento (Torres, 2014). A gestão de incidentes (GI) surge como um importante componente da SI e pode ser estabelecida para o desenvolvimento e apoio às habilidades supracitadas (Cichonski, et al., 2012.).

O estabelecimento da capacidade de GI requer a definição de procedimentos, atribuição de funções e responsabilidades, infraestrutura, ferramentas e materiais de apoio adequados e equipe qualificada e treinada para a realização de um trabalho consistente, confiável, de alta qualidade e capaz de ser repetível (Killcrece, 2005). Quanto aos procedimentos, cabe destacar que o processo de GI envolve atividades relacionadas a coordenação, suporte, avaliação de incidentes e tratamento de incidentes, que por sua vez é composto por diversas fases: detecção, triagem e resposta. Para desempenhar tais atividades são exigidos conhecimentos de múltiplas áreas, ademais muitas das atividades envolvidas são de natureza não determinística (Mundie e Ruefle, 2012). Eis, portanto, um contexto multidisciplinar e complexo, que resulta em muitos desafios. Entre os desafios, destacam-se a capacitação de profissionais para o tratamento de incidentes a partir de conhecimentos prévios do domínio da GI. (Torres, 2014) As dificuldades relacionadas associam-se a

fatores como a difícil formalização dos conhecimentos tácitos dos envolvidos (Mundie e Ruefle, 2012), um desafio da área que não limita-se ao domínio da GI.

O trabalho propõe o desenvolvimento de uma ontologia para o domínio da gestão de incidentes de SI baseada na norma ISO/IEC 27035:2011 e objetiva auxiliar na capacitação de equipes de reposta a incidentes. São referências normativas adicionais as normas ISO/IEC 27001:2013, ISO/IEC 27002:2005 e ISO/IEC 27005:2008.

2. Trabalhos Relacionados

Os trabalhos a seguir relacionados à ontologias e à GI são apresentados e comparados ao proposto (Tabela 1). São utilizados os seguintes critérios de comparação: (i) classificação (vocabulário, tipo de ontologia, taxonomia etc); (ii) referências; (iii) registro de metodologia para construção da ontologia (iv) registro de metodologia para avaliação da ontologia e (v) disponibilidade para a comunidade/publicação.

- Martimiano e Moreira (2005): ontologia que objetiva permitir a correlação de incidentes de SI de diferentes fontes e facilitar a gestão do conhecimento propondo vocabulário único de termos e relações fundamentado no documento *Taxonomy of the Computer Security Incident (TCSI)* de Howard e Longstaff. Não exemplifica a referida correlação.
- Blackwell (2010): ontologia focada na etapa de análise de incidentes e fundamentada no *TCSI* - por ater-se a este, desatende o atual cenário da GI.
- Mudie e Ruefle (2012): *Body of Knowlodge* para GI que visa a possibilitar uma definição da área de conhecimento; padronizar competências, vocabulários e processos; facilitar a criação de um repositório; orientar a descrição de requisitos, formações e competências exigidas a profissionais; e propiciar análise/melhoria do processo de GI nas organizações. Baseia-se em dez documentos da área de SI e/ou GI. Não elege uma metodologia específica para construção, mas lista suas etapas.

Tabela 1. Comparativo trabalho proposto e trabalhos relacionados

Trabalhos	Classificação	Ref.ISO/IEC 27035/2011	Metodologia construção	Metodologia avaliação	Publicada
Martimiano e Moreira(2005)	Não especificado	Não	Sim	Não	Não
Blackwell (2010)	Não especificado	Não	Não	Não	Não
Mudie e Ruefle (2012)	Body Of Knowlodge	Não	Não se aplica	Não se aplica	Não
SIMO	Ontologia de Dominio	Sim	Sim	Sim	Sim

Dos trabalhos relacionados, a ontologia proposta distingue-se quanto à referência primária, a norma ISO/IEC 27035:2011, à disponibilidade aos usuários via Web Protégé, aos métodos avaliativos e, sobretudo, ao objetivo de auxiliar a capacitação de equipes de GI.

3. Trabalho Proposto - Ontologia

De acordo Rautenberg et al. (2008), há várias metodologias para o desenvolvimento de ontologias, não há consenso quanto a um padrão; recomenda-se, a combinação de

metodologias. A partir das metodologias de Noy e Mcguinness (2001), Sure et al., (2004) e Bouiadjra e Benslimane (2011), elaborou-se a metodologia a seguir, conforme figura 1.

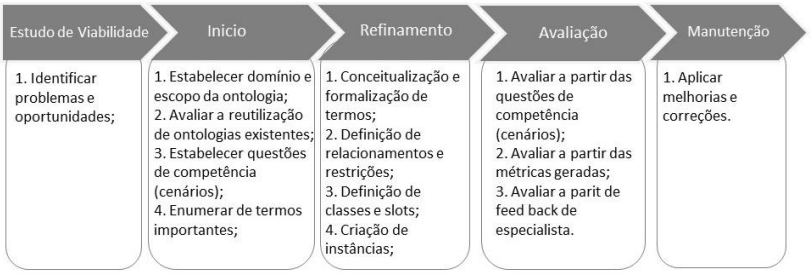


Figura 1. Metodologia Utilizada

Conforme Noy e Mcguinness (2001), a ontologia deve responder a cinco questões: Qual domínio a ontologia abrange? Representação do conhecimento de GI necessários a equipes de resposta; A ontologia será utiliza para quê? Auxiliar na capacitação de equipes de resposta; Qual questão a ontologia deve ser capaz de responder? Deve ser capaz de fornecer informações relacionadas a um incidente de SI, como sua categoria, classificação, causa ou ameaça, ativos envolvidos, impacto do incidente, ações de resposta, vulnerabilidade(s) e atacante ou perpetrador.

A partir da enumeração de termos, mapearam-se os termos e suas relações. Após o refinamento, consolidou-se uma visão da ontologia proposta, conforme figura 2. Cada uma das classes definidas apresenta subclasses, propriedades e instâncias considerando as normas utilizadas como referências.

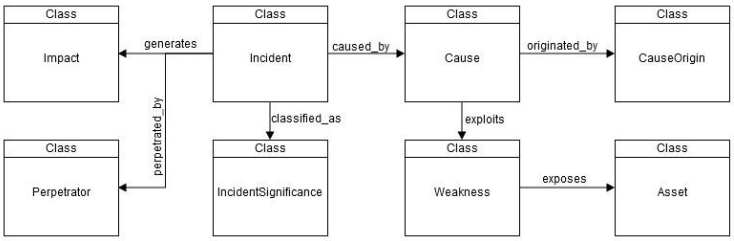


Figura 2. Mapeamento ontologia com classes e propriedades de objetos

As figuras 3 e 4 descrevem a ontologia proposta a partir da classe Incidente, expondo algumas das propriedades de dados e propriedades de objetos. A ontologia conta ainda com outras 11 propriedades de objetos e 11 propriedades de dados (4 de primeiro nível e 7 de segundo nível), totalizando 16 e 25 respectivamente.

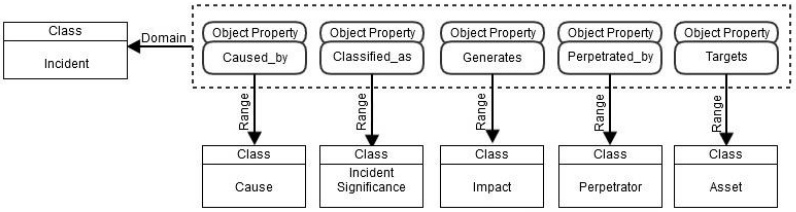


Figura 3. Descrição Ontologia Parcial - Classe Incidente e Objetos

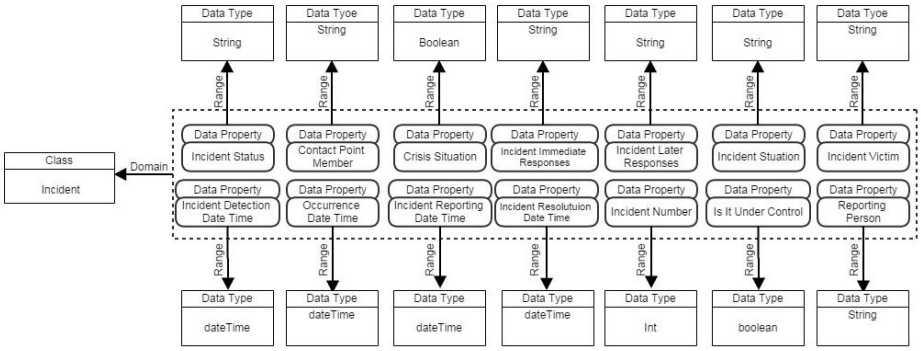


Figura 4. Descrição Ontologia Parcial - Classe Incidente e Dados

4. Considerações Parciais

Nesse momento, objetiva-se a criação de cenários de incidentes adicionais para avaliar a competência da ontologia, além da submissão do trabalho à avaliação de profissionais de equipes de resposta a incidentes, no intento de verificar sua efetividade. A geração de um protótipo de software para GI a partir da ontologia também é uma proposta de trabalho futuro. A ontologia está disponível para consulta e utilização em http://webprotege.stanford.edu/ pelo nome SIMO.

Referências

Blackwell, C.(2010) "A security ontology for incident analysis", Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence research, p. 1-4.

Bouiadjra, A., B., E Benslimane, S. (2011) "FOEval: Full Ontology Evaluation -Model and Perspectives", IEEE, Tokushima, p. 464-468

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012) "Computer Security Incident Handling Guide." Recommendations of the NIST, Gaithersburg,

ISO/IEC 27035 (2011) "Information technology – Security techniques – Information security incident management".

Killcrece, G. (2005) "Incident Management", SEI CMU, Pittsburgh, Pennsylvania.

Martimiano, L. A. F.; Moreira, E. Dos S. (2005) "An OWL-based Security Incident Ontology".

Martimiano, L. A. F.; Moreira, E. Dos S. (2006) "The Evaluation Process of a Computer Security Incident Ontology", Ribeirão Preto.

Mccarthy, N.K. (2014) "Resposta a Incidentes de Segurança em Computadores: Planos para Proteção de Informação em Risco", Porto Alegre, Bookman.

Mundie, D. A., E Ruefle, R. (2012) "Building an Incident Management Body of Knowledge", In: ARES, Washington , p. 507–513

Noy, F. N., E Mcguinness, D. L. (2001) "Ontology development 101: a guide to create your first ontology"

Rautenberg, S., Todesco, J., Steil, A. E Gauthier, F. (2008) "Uma Metodologia para o Desenvolvimento de Ontologias", Guarpuava, Paraná, v. 10, n. 2, p. 237-262

Sure, Y., Staab, S. E Studer, R. (2004) "On-To-Knowledge Methodology". Nova York, Springer.

Torres, A. (2014) "Incident Response: How to Fight Back", Survey Incident Response 2014, SANS Institute.

S-MOVL: Protegendo Sistemas Computacionais contra Ataques de Violação de Memória por meio de Instruções em Hardware

Antonio L. Maia Neto, Omar P. Vilela Neto, Fernando M. Q. Pereira, Leonardo B. Oliveira

[1] Departamento de Ciência da Computação (DCC)
Universidade Federal de Minas Gerais (UFMG) - Belo Horizonte, MG - Brazil

{lemosmaia,omar,fernando,leob}@dcc.ufmg.br

Abstract. *The C programming language does not prevent ouf-of-bounds memory access and thus gives room to attacks such as buffer overflow and buffer overread. There exist several techniques to secure C programs. Nevertheless, these methods are usually implemented via software and therefore tend to slow down programs and frequently compromise performance of applications. This work aims at presenting a hardware solution able to check bounds efficiently.*

Resumo. *A linguagem C não verifica limites de arranjo e abre brechas para ataques de violação de memória, tais como buffer overflow e o buffer overread. A maioria das propostas para adicionar essa funcionalidade à linguagem são implementadas em software, estratégia que prejudica o desempenho de aplicações. Neste trabalho apresentamos uma solução em hardware capaz de realizar a verificação de limites eficientemente.*

1. Introdução

A linguagem C é uma das mais empregadas em meio à comunidade de programadores e foi concebida com o foco na eficiência, permitindo que as aplicações executem numa velocidade compatível com os recursos computacionais muitas vezes escassos. No entanto, um preço alto é pago nesta busca por eficiência. C, por exemplo, não realiza a Verificação de Limites de Arranjo (*Array-Bounds Check* – ABC) automaticamente. Ao invés disso, a linguagem deixa a cargo do programador inserir ABCs quando este achar necessário.

O resultado dessa estratégia é que uma grande gama de programas escritos em C estão sujeitos a ataques que acessam memória para além de limites, tais como Estouro de Arranjo (*Buffer Overflow* – BOF) e Leitura pós Arranjo (*Buffer Overread* – BOR). Para ilustrar a capacidade destrutiva desses ataques, pode-se se citar o *worm* Morris[1], que nos idos de 80 abalou a Internet ao explorar uma vulnerabilidade de BOF causando um ataque de DoS sem precedentes; e, recentemente, o Heartbleed[2], que no ano passado deixou a comunidade de segurança digital em polvorosa, ao explorar a vulnerabilidade de BOR e, assim, furtar dados sigilosos de programas que utilizavam a biblioteca OpenSSL.

Evidentemente, ao longo dos anos diversas propostas surgiram com o intuito de proteger programas escritos em C ([Dhurjati et al. 2006], por exemplo). Grosso modo,

[1] http://en.wikipedia.org/wiki/Morris_worm
[2] http://en.wikipedia.org/wiki/Heartbleed

os trabalhos existentes analisam programas, identificando os locais das vulnerabilidades e, posteriormente, inserindo ABCs nesses trechos de código. Em tese, tais propostas são capazes de contornar a questão da verificação de limites. Todavia, na prática, elas acabam sendo ineficientes. O problema está no fato de que a ABC feita em software acarreta alta sobrecarga (*overhead*).

O objetivo deste artigo é conceber instruções em linguagem de máquina que realizem, de forma segura, o transporte de dados da memória para registradores e vice-versa. Desta forma, deixamos a cargo do hardware – mais eficiente que o software – checar os limites de arranjos. Nossa proposta, chamada de S-MOVL, cria versões seguras das instruções de leitura (*load*) e escrita (*store*) em memória, onde os limites inferior e superior do arranjo manipulado são verificados antes da conclusão do acesso à memória.

2. Trabalhos Relacionados

Há na literatura uma infinidade de técnicas para proteger sistemas computacionais. Por motivos de restrição de espaço, vamos nos concentrar àquelas relativas a ataques de Violação de Memória e que, além disso, lancem mão de modificações em hardware para melhorar os níveis de segurança e de desempenho das aplicações.

Grande parte das propostas de modificações em hardware para conter ataques de Violação de Memória tem como principal objetivo proteger dados de controle, endereços de retorno e ponteiros de funções, de ataques de desvio de fluxo [Piromsopa and Enbody 2006]. Esse tipo de abordagem, ao atacar especificamente problemas de BOF, não impede que um sistema computacional esteja vulnerável a ataques de BOR, por exemplo. Nossa solução visa prover uma forma de defesa que abrange todos os tipos de ataques de Violação de Memória.

Assim como a nossa solução, existem trabalhos que, através de modificações em hardware, objetivam mitigar os ataques de Violação de Memória em sua totalidade. Dentre essas propostas está o conjunto de Extensões de Proteção de Memória (*Memory Protection Extensions* – MPX) adotadas na nova geração dos processadores Intel [Intel Corporation 2013]. Esse conjunto de extensões consiste em uma série de novas (oito) instruções de gerenciamento e verificação de limites de arranjo. Considerando o uso da extensão MPX, o processo de acesso seguro a uma posição de memória é: **i)** carregar os limites superior e inferior de um arranjo em registradores especiais – instrução `BNDMK`; **ii)** verificar os limites superior e inferior – instruções `BNDCL` e `BNDCU`, respectivamente; e, finalmente, **iii)** concluir o acesso (escrita ou leitura) à memória – instruções tradicionais de `mov`.

A principal diferença entre nossa solução e o ferramental fornecido em MPX é a possibilidade de, no nosso caso, concluir a verificação de limites e o acesso à memória por meio de uma única instrução – `srmmovl` para escrita e `smrmovl` para leitura. Outro ponto que difere as abordagens é com relação às mudanças em hardware. Enquanto MPX cria novos registradores internos no processador, nós nos atemos aos registradores de propósito geral já existentes.

3. S-MOVL

Optamos por projetar nossa solução sobre a arquitetura Y86. A arquitetura Y86 foi inspirada na arquitetura IA32 e é atualmente uma das mais empregadas pela academia. Embora

menor que o da IA32, o ISA da Y86 permite a execução de programas suficientemente complexos para se avaliar desempenhos. Ademais, informações detalhadas sobre o projeto da arquitetura estão públicas [Bryant and David Richard 2003].

As nossas instruções – da forma **`s-movl rA, rB, rX, rY`** – consistem em versões seguras das instruções de leitura e escrita na memória que, antes de concluir um acesso, validam os limites do arranjo. A verificação dos limites é feita por comparações, que, no caso do Y86, são resultado da avaliação de subtrações. Portanto, a primeira comparação foi baseada na instrução já existente `subl rA, rB`, onde a operação `rB - rA` é computada no estágio *Execute* do *pipeline*. Logo, podemos utilizar o registrador `rB` para armazenar o endereço do limite superior do arranjo e o registrador `rA` para armazenar o endereço de acesso. Para a segunda comparação, verificação do limite inferior, é importante notar que o estágio *Execute* é incapaz de computar mais de uma operação por ciclo. Assim, com o intuito de não gerar uma sobrecarga de tempo à nova instrução, adicionamos um novo componente de hardware a esta etapa que, em paralelo, calcula a diferença entre o endereço de acesso e o limite inferior do arranjo. E então, o registrador `rX` armazena o limite inferior do arranjo e o novo componente computa a operação `rA - rX`.

Optamos por seguir a convenção da linguagem C considerando que o sinal de controle que valida o acesso à memória quanto ao limite superior do arranjo é verdadeiro se o resultado da primeira subtração for estritamente maior que zero. Analogamente, em relação ao limite inferior do arranjo, o acesso à memória é garantido se o resultado da segunda subtração for maior ou igual a zero. A tentativa de acesso fora dos limites do arranjo, detectada por uma das comparações descritas acima, deve, obrigatoriamente, ser sinalizada por uma exceção de hardware.

Os estágios de execução ao longo do *pipeline* da arquitetura Y86 modificada são exibidos na Tabela 1. Nela, as operações adicionadas/modificadas foram destacadas.

Stage	**`srmmovl rA,rB,rX,rY`**	**`smrmovl rA,rB,rX,rY`**
Fetch	$icode : ifun \leftarrow M_1[PC]$	$icode : ifun \leftarrow M_1[PC]$
	$rA : rB \leftarrow M_1[PC+1]$	$rA : rB \leftarrow M_1[PC+1]$
	$\mathbf{rX : rY \leftarrow M_1[PC+2]}$	$\mathbf{rX : rY \leftarrow M_1[PC+2]}$
	$valP \leftarrow PC+3$	$valP \leftarrow PC+3$
	$PC \leftarrow valP$	$PC \leftarrow valP$
Decode	$valA \leftarrow R[rA]$	$valA \leftarrow R[rA]$
	$valB \leftarrow R[rB]$	$valB \leftarrow R[rB]$
	$\mathbf{valX \leftarrow R[rX]}$	$\mathbf{valX \leftarrow R[rX]}$
	$\mathbf{valY \leftarrow R[rY]}$	
Execute	$\mathbf{if(valB - valA > 0) \rightarrow exception}$	$\mathbf{if(valB - valA > 0) \rightarrow exception}$
	$\mathbf{if(valA - valX \geq 0) \rightarrow exception}$	$\mathbf{if(valA - valX \geq 0) \rightarrow exception}$
Memory	$\mathbf{M_4[valA] \leftarrow valY}$	$\mathbf{valM \leftarrow M_4[valA]}$
Write back		$\mathbf{R[rY] \leftarrow valM}$

Tabela 1. Estágios do *pipeline* do Y86 para das instruções seguras.

4. Resultados

Nossa solução foi avaliada sobre uma versão ligeiramente modificada do programa *bubblesort* pertencente ao *benchmark* Stanford. Foi considerado apenas um arranjo de tamanho igual a 400 inicializado em ordem decrescente para, então, ser ordenado. Foram criadas as seguintes versões do programa: **original** – os arranjos são acessados sem quaisquer ABCs; **baseline** – a cada acesso a arranjos, tanto atribuições quanto leituras, é utilizado um ABC via software; **S-MOVL** – o código em linguagem de montagem da versão

original é analisado de forma a identificar as instruções de *load* ou *store* que acessam arranjos. Essa instruções são, então, trocadas por uma sequência equivalente de instruções concluída por uma instrução S-MOVL.

Cada uma das versões foi executada no simulador Y86 descrito em [Bryant and David Richard 2003] e disponível online[3], fornecendo o total de número de ciclos e instruções durante cada execução. Os resultados, sintetizados na Tabela 2, mostram que os ABCs em software causaram uma sobrecarga de 123,86% no total de instruções e 149,39% em número de ciclos. Já a versão em hardware reduziu as sobrecargas para 58.29% e 44,27%, respectivamente. Esse resultado mostra que S-MOVL precisou de 57,85% ciclos a menos que o *baseline* para executar.

bubblesort	**Números**			**Sobrecarga**	
	Instruções	**Ciclos**	**CPI**	**Instruções**	**Ciclos**
original	3293834	4337245	1.32		
baseline	7373623	10816629	1.47	123.86%	149.39%
S-MOVL	5213834	6257245	1.20	58.29%	44.27%

Tabela 2. Resultados das simulações das 3 versões do programa *bubbesort*

5. Conclusão

As propostas de automatização de ABCs são, em sua maioria, técnicas de instrumentação via software, que tendem a prejudicar o desempenho dos sistemas já que aumentam significativamente a quantidade de código dos programas. O objetivo deste trabalho foi propor uma solução eficiente de verificação de limites em hardware, através de instruções seguras de leitura e escrita na memória para a arquitetura Y86. O resultado das comparações da nossa solução frente à estratégia em software apontaram melhoras de cerca de 58% no desempenho dos programas analisados.

Como continuidade do projeto pretendemos avaliar nossa solução em relação a outras estratégias em hardware, principalmente a extensão MPX.

Referências

Bryant, R. and David Richard, O. (2003). *Computer systems: a programmer's perspective*. Prentice Hall.

Dhurjati, D., Kowshik, S., and Adve, V. (2006). SAFECode: enforcing alias analysis for weakly typed languages. In *ACM SIGPLAN conference on Programming language design and implementation - (PLDI '06)*, pages 144–157.

Intel Corporation (2013). Intel Architecture Instruction Set Extensions Programing Reference. `http://download-software.intel.com/sites/default/files/319433-015.pdf`.

Piromsopa, K. and Enbody, R. J. (2006). Secure bit: Transparent, hardware buffer-overflow protection. *IEEE Transactions on Dependable and Secure Computing*, 3(4):365–376.

[3]`http://csapp.cs.cmu.edu/public/labs.html`

Arquitetura de monitoramento para Security-SLA em Nuvem Computacional do tipo SaaS

Carlos Alberto da Silva[1], **Paulo Lício de Geus**[1]

[1] Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
{ beto, paulo}@lasca.ic.unicamp.br

Abstract. *Cloud Computing has introduced new technology and architectures that changed enterprise computing. In particular, when hiring a service in the cloud, an important aspect is how security policies will be applied in this environment characterized by both virtualization and large-scale multi-tenancy service. Security metrics can be seen as tools to provide information about the status of the environment. Aimed at improving security in cloud, this paper presents an architecture for security monitoring based on Security-SLA for SaaS services.*

Resumo. *Nuvem Computacional introduziu novas tecnologias e arquiteturas que modificaram a computação empresarial. Em particular, ao contratar um serviço na nuvem, um aspecto importante é a forma como as políticas de segurança serão aplicadas neste ambiente caracterizado pela virtualização e serviço de multilocação em grande escala. Métricas de segurança podem ser vistas como ferramentas para fornecer informações sobre o estado deste ambiente. Visando a melhoria da segurança em nuvens, este artigo apresenta uma arquitetura para monitoramento de segurança baseado em Security-SLA para serviços SaaS.*

Palavras chaves: Métricas de Segurança, Security-SLA, Segurança em Nuvem.

1. Introdução

Nuvem Computacional define-se como um modelo para permitir acesso fácil, a rede sob demanda para um conjunto compartilhado de recursos configuráveis de computação como: redes, servidores, armazenamento, aplicações e serviços, que podem ser rapidamente provisionados e liberados com um esforço mínimo de gerenciamento ou interação com o provedor destes serviços. Os três tipos principais de serviços oferecidos por provedores de nuvem computacional são: Infraestrutura-como-um-serviço (IaaS), Plataforma-como-um-Serviço (PaaS) e Software-como-um-Serviço (SaaS).

Os potenciais clientes de nuvem percebem uma ausência de transparência e uma relativa falta de controles, quando comparado com os modelos tradicionais [Pearson 2013].

As qualidades especificadas em um Security-SLA podem ser classificadas em mensurável e não mensurável. As qualidades mensuráveis são medidas automaticamente por meio de métricas, e as qualidades não mensuráveis não permitem uma medição automática, ou através de um método que resulta em um valor único. As qualidades encontradas em serviços de TI são: (a) Mensuráveis: precisão, disponibilidade, capacidade, custo,

latência, tempo de provisionamento, confiabilidade e escalabilidade; (b) Não mensurável: interoperabilidade, modificabilidade e segurança.

Diante deste cenário, este trabalho apresenta uma solução de monitoramento que acompanha os acordos de nível de serviço de segurança, Security-SLA, utilizando um sistema de monitoramento de segurança que baseia-se em uma hierarquia de métricas de seguranças para Infraestrutura e Serviço contratado, e discute também a utilização de escala de valores para tratar o problema de aferir qualidades não mensuráveis.

2. Trabalhos Relacionados

Atualmente, as soluções comerciais de monitoramento permitem apenas o monitoramento de informações básicas como carga de CPU, uso de espaço de armazenamento e tráfego de rede, tais como: (i) plataforma AWS da *Amazon* oferece o *CloudWatch*, um sistema de monitoramento oferecido como um serviço para o controle de recursos; (ii) *Microsoft Windows Azure* possui o *Azure Fabric Controller* que monitora e gerencia os recursos e serviços dos servidores; (iii) *Google App Engine* oferece um conjunto de APIs que permitem a utilização de soluções de monitoramento como o *CloudStatus*; (iv) assim como as soluções de nuvens de código aberto como: *Eucalyptus*, *OpenNebula* e *OpenStack*.

Na área acadêmica, o monitoramento da nuvem computacional apresenta poucos resultados concretos [Shao et al. 2010], e os sistemas de monitoramento são voltados para o monitoramento de aplicações específicas, e não estão associados ao acompanhamento dos acordos de Security-SLA.

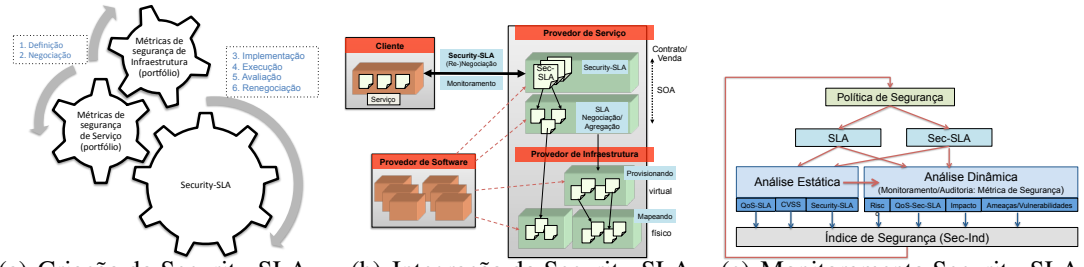
Em Emeakaroha et al. [Emeakaroha et al. 2010] apresentam a solução de monitoramento baseada no protocolo SNMP e métricas de desempenho, que são posteriormente utilizadas por um módulo de detecção de violação do acordo de SLA.

3. Solução de Monitoramento

A solução de monitoramento proposta tem o objetivo de acompanhar o cumprimento de acordos de Security-SLA para nuvem SaaS, e é dividida em: [1a parte] o monitoramento ocorre nos dispositivos de infraestrutura da nuvem, como: *firewalls*, roteadores, comutadores, *proxies*, etc.; [2a parte] o monitoramento ocorre sobre o serviço contratado utilizando técnicas de monitoramento de caixa-preta ou a introspecção da Maquina Virtual (VM), que permitem a coleta de informações sem a necessidade de instalação de ferramentas no sistema operacional da VM. O mecanismo de Introspecção da VM da biblioteca LibVMI permite o acesso aos aspectos da VM no *hypervisor* como: memória, utilização do processador, entradas e saídas de dados [VMITools 2013].

Esta solução apresenta o acordo de Security-SLA através de uma linguagem XML que permite especificar os serviços e as políticas de segurança que serão monitoradas através de métricas. Apesar de sua arquitetura ser voltada ao controle de acordos de Security-SLA, a solução é flexível o suficiente para permitir o monitoramento de outros tipos de requesitos, como por exemplo, qualidade do serviço (QoS), risco e impacto.

A Figura 1 apresenta a arquitetura de monitoramento do Security-SLA. Na figura 1(a) descreve a criação do Security-SLA a partir dos portfólios de métricas de segurança, onde cada métrica está associada a um Objetivo de Nível de Serviço (SLO). A Figura 1(b) descreve como o Security-SLA irá interagir com a infraestrutura física e virtual da



(a) Criação do Security-SLA (b) Integração do Security-SLA (c) Monitoramento Security-SLA

Figura 1. Arquitetura de monitoramento do Security-SLA

nuvem. E a Figura 1(c) representa o processo de monitoramento, onde a política de segurança define o SLA e Security-SLA. Na sequência, o processo de monitoramento das métricas é dividido em duas etapas: (1) Análise Estática: especifica como as métricas serão aferidas e os valores de comparação para: (i) SLA e Security-SLA; (ii) as qualidades do serviço do SLA (QoS-SLA); (iii) as vulnerabilidades registradas no *National Vulnerability Database* (NVD) para o tipo de serviço, identificando o valor de risco e impacto; (2) Análise Dinâmica: é executado o processo de aferir as métricas definidas na fase anterior, comparando valores de SLOs com valores aferidos (MA). Neste modelo, cada cláusula do acordo (SLO) está associado a uma métrica de segurança, e assume valores no intervalo de [0-4], e este intervalo representa os níveis de segurança permitidos [Crítico, Alto, Médio, Baixo, Zero]. Resultando ao final do processo um índice de segurança (Sec-Ind).

O processo de monitoramento depende do tipo de serviço sendo contratado, onde as ameaças e vulnerabilidades são identificadas para este perfil de serviço, e calcula-se o risco e impacto das operações executadas sobre o serviço usando como referência no NVD do *Common Vulnerability Scoring System* (CVSS).

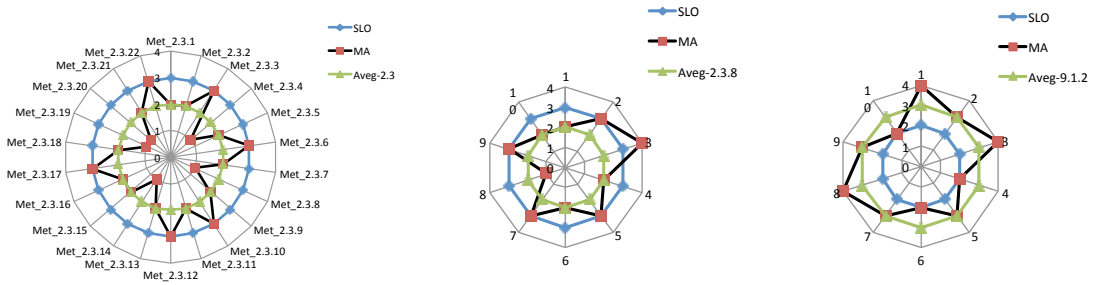
Para validar as métricas de segurança coletadas, a arquitetura de monitoramento realiza duas etapas: (1) Os valores aferidos por métricas de segurança entre [0-4] são classificados como: verdadeiros-positivo (VP), falsos-positivo (FP), verdadeiros-negativo (VN) e falsos-negativo (FN); (2) Indicadores de validação são calculados para o modelo: (i) *Precision*: $P = \frac{VP}{VP+FP}$, indica o percentual de eventos corretamente classificados como incidente entre aqueles que foram classificados como incidentes; (ii) *Recall*: $R = \frac{VP}{VP+FN}$, indica a percentual de eventos corretamente classificados como incidentes entre todos os eventos que são efetivamente incidentes; (iii) *F-measure*: $F = \frac{2 \times P \times R}{P+R}$, é a média harmônica entre Precisão e Recall; (iv) *Accuracy*: $A = \frac{VP+VN}{VP+VN+FP+FN}$, indica o percentual de eventos corretamente classificados.

Analisando os valores dos Indicadores de validação, é determinado o grau de confiabilidade dos valores coletados pelas métricas de segurança.

3.1. Estudo de Caso

Um estudo de caso foi desenvolvido e testado em um ambiente de nuvem computacional com base no *OpenNebula*, em uma máquina com *Gentoo Linux*, *hypervisor* KVM e banco de dados *PostgreSQL*. O sistema em teste é responsável pela gestão de recursos humanos em uma universidade e possui cerca de 400 tabelas, 200 usuários e 5 administradores.

A Figura 2 apresenta o resultado do monitoramento do Security-SLA, onde a Figura 2(a)



(a) Métrica *Firewall* (b) Sub-métrica *Packet filtering* (c) *Insecure user account*

Figura 2. Resultado do monitoramento do Security-SLA

é a métrica *Firewall* ($Met_{2.3}$), a Figura 2(b) é a sub-métrica *Packet filtering* ($Met_{2.3.8}$) e a Figura 2(c) é a sub-métrica *Insecure user account* ($Met_{9.1.2}$) do *PostgreSQL*. E os parametros: SLO é o valor acordado no Security-SLA, MA é o valor aferido da métrica e *Aveg* é a média de MA nas 10 amostras coletadas das figuras 2(b) e 2(c). As Figuras 2(b) e 2(c) apresentam valores aferidos menores que o contratado (violação de acordo).

4. Conclusão e Trabalhos Futuro

Nós apresentamos contribuições substanciais para uma arquitetura de monitoramento de Security-SLA através de métricas de segurança. O intervalo de valores de [0-4] para cada SLO ou métricas de segurança apresenta-se como uma nova abordagem para tratar os valores não mensuráveis ou intangíveis do ambiente de nuvem computacional.

Como trabalho futuro, consideramos o desenvolvimento de mapeamentos dinâmicos dos SLOs entre Security-SLA e os modelos de serviços existentes através de uma pré-análise de *sockets* (número IP + porta). E automatizar o processo de contramedidas para minimizar as violações dos Security-SLA e maximizar o nível de segurança do ambiente de nuvem computacional.

Agradecimentos

Os autores agradecem o apoio financeiro da CAPES e Fundect (Processo #23/200.308/2009).

Referências

Emeakaroha, V. C., Calheiros, R. N., Netto, M. A. S., Brandic, I., and Rose, C. A. F. D. R. A. F. D. (2010). Desvi: An architecture for detecting sla violations in cloud computing infrastructures. In *2nd International ICST Conference on Cloud Computing*, CloudComp 2010.

Pearson, S. (2013). Toward accountability in the cloud. *Jornal IEEE Cloud Computing - Especial Edition: Securing the Cloud*, 1(1):6–10.

Shao, J., Wei, H., and Mei, H. (2010). A runtime model based monitoring approach for cloud. *IEEE 3rd International Conference on Cloud Computing - CLOUD'10*, page 313–320.

VMITools (2013). Virtual machine introspection tools. Technical report, Available in `https://code.google.com/p/vmitools/`. Acessado em 5 julho de 2014.

Detecção Estática e Consistente de Potenciais Estouros de Arranjos

Bruno Rodrigues Silva[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

brunors@dcc.ufmg.br

Resumo. *Estouros de arranjos, uma vulnerabilidade de software bastante conhecida na literatura especializada em segurança, são frequentemente utilizados por adversários que têm o objetivo de corromper o fluxo de controle do programa. Isto é possível em linguagens fracamente tipadas como C e C++ onde os acessos à arranjos não são verificados. Uma possível solução é a inserção de código para verificação de todos os acessos à arranjos de forma a evitar aqueles fora dos limites. Entretanto o custo em tempo de execução seria proibitivo. Este trabalho propõe a detecção de potenciais estouros de arranjos via análise estática de código, o que permitiria ao desenvolvedor, a inserção de código de verificação apenas nos traços estáticos possivelmente vulneráveis. Verificou-se que cerca de 41% do código fonte dos benchmarks SPEC CPUINT 2006 estão vulneráveis à este tipo de ataque.*

1. Introdução

Um estouro de arranjo acontece quando este é preenchido com dados que ultrapassam os seus limites. Isso pode acontecer de forma proposital ou não, principalmente em linguagens fracamente tipadas e bastante utilizadas como C e C++ que não fazem verificação dos acessos ao arranjo. O que possibilita a existência de uma grande quantidade de *worms* e vírus que contaminaram e contaminam milhões de dispositivos computacionais em todo o mundo. O estouro de arranjo pode sobreescrever os valores de variáveis locais e endereço de retorno de função, o que compromete todo o fluxo de dados e controle.

Usuários maliciosos de posse do código fonte podem manipular os dados de entrada que são públicos a fim de estourar um arranjo e sobreescrever o endereço de retorno de uma função com informações que direcionem o fluxo de controle para funções de sistema tais como *telnet* e *shell* com os mesmos privilégios de sistema do programa atacado. Este redirecionamento pode causar desde a interrupção do serviço até a obtenção do controle total do sistema.

Uma solução empregada por muitos compiladores é a inserção de canários, que são valores aleatórios inseridos antes do valor de retorno de uma função. Além disso, é inserido uma pequena seção de código de verificação antes do retorno, capaz de gerar uma exceção e encerrar o programa caso o canário tenha sido modificado. Portanto, caso um adversário tente sobreescrever o valor de retorno, inevitavelmente ele também sobrescreverá o canário que por sua vez será detectado no momento do retorno da função.

Entretanto, mesmo funções protegidas por canários, estão vulneráveis ao ataque de estouro de arranjo [Maffra et al. 2013]. Isto acontece porque algumas variáveis locais

podem ser alojadas na pilha da função após o espaço alocado para o arranjo e portanto, elas podem ser sobreescritas em um possível estouro. Um adversário pode, através de um estudo cauteloso do fluxo de dados/controle da função, escolher quais valores tais variáveis receberão e assim comprometer a execução de todo o programa.

O trabalho de Quadros *et. al*. [Quadros et al. 2012] demonstra como um adversário pode assumir o controle total de um sistema executando Ubuntu Linux, por meio de um ataque de estouro de arranjo. Em [Maffra et al. 2013] Maffra *et al.* propõem uma análise estática de código para a detecção de vulnerabilidade de ataque por estouro de arranjo em código compilado com canários. Entretanto, por não considerar as dependências de controle entre as variáveis e os predicados de instruções de desvio, definidas na próxima Seção, sua análise não é consistente. Uma análise é considerada consistente se ela não gera falsos negativos, isto é, a análise não pode reportar a inexistência de vulnerabilidade, quando na verdade ela existe.

Nesse trabalho, propõe-se uma análise estática de código capaz de detectar de forma consistente os possíveis traços de código vulneráveis ao ataque de estouro de arranjo. Tal análise foi implementada como um módulo para o compilador LLVM [Lattner and Adve 2004] e executada sobre o conjunto de benchmarks SPEC CPU INT 2006[1].

2. Análise Estática

Um grafo de dependências $G = (V, E)$ tal como proposto por Ferrante [Ferrante et al. 1987] é utilizado na implementação dessa análise estática para detecção consistente de vulnerabilidade de estouro de arranjo. O conjunto V de vértices contém variáveis mapeadas em registradores, conjuntos de posições de memória[2] e operações. Já o conjunto E representa as relações de dependências de dados e de controle entre os vértices contidos em V. Para cada variável u definida com a informação contida em outra variável v, temos uma aresta direcional conectando v à u, capturando assim a relação de dependência de dados entre essas duas variáveis.

Por outro lado, informação também flui implicitamente de um predicado p, que controla um teste condicional, para toda variável atribuída no escopo desse teste. Por exemplo, a sequência `p = (a > b); v = p?0:1;` determina uma dependência de controle de p para v e portanto uma aresta direcional conectando p à v também é inserida no conjunto E do grafo de dependências. Não considerar os fluxos implícitos em uma análise de fluxo de informação é um erro, conforme descrito em Silva *et. al* [Silva 2013].

O algoritmo para construção do grafo de dependências pode ser encontrado em Silva [Silva 2013] e uma versão linear sobre o número de variáveis do programa pode se encontrada em [Silva 2014]. Portanto, por limitações de espaço, ele não será descrito nesse texto. Uma vez que o grafo de dependência tenha sido construído, é correto afirmar que todas as relações de dependência de controle e de dados entre as variáveis e posições de memória do programa estão devidamente presentes nesta nova representação intermediária. Tal representação é então utilizada na busca por traços estáticos de código que possibilitem estouro de arranjo, isto é, caminhos contaminados no fluxo de informação.

[1]Código fonte disponível na página do projeto E-CoSoC - https://code.google.com/p/ecosoc/

[2]Uma análise de ponteiros é utilizada na construção de tais conjuntos

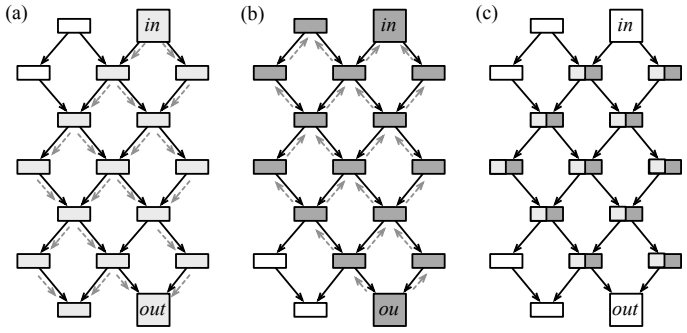


Figura 1. a) Busca em profundidade a partir de um vértice do conjunto de entradas. b) Busca em profundidade invertida à partir de um vértice do conjunto de sorvedouros. c) Vértices marcados em ambas as buscas

A busca por caminhos contaminados é baseada na localização de arranjos que são dependentes de entradas que um adversário pode manipular. Para tanto, é definido um conjunto G_e de entradas públicas que, seguindo o estudo realizado por Mafra *et. al* [Maffra et al. 2013], será constituído pelos seguintes elementos:

1. os argumentos do método *main*, isto é, as variáveis *argc* e *argv*;
2. o resultado retornado por funções externas de entrada tais como *scanf*, *fgets*, *read*;
3. ponteiros passados como argumento de funções externas;

Um conjunto de sorvedouros G_s que contém todos as ponteiros para início de arranjos locais também é construído. Em seguida é realizada uma busca em profundidade a partir de cada vértice do conjunto G_e, onde cada vértice alcançável é marcado. Uma segunda busca é realizada a partir de cada vértice do conjunto G_s seguindo o sentido contrário das arestas. Novamente todo os vértices alcançáveis por cada sorvedouro são marcados. A Figura 1a) esboça a busca e marcação de vértices a partir de um vértice de entrada. A busca invertida a partir de um sorvedouro pode ser visualizada na Figura 1b). Finalmente a Figura 1c) exibe os vértices presentes na interseção das duas buscas. Estes são, portanto, os possíveis caminhos que conectam uma entrada pública que pode ser manipulada pelo adversário, à uma acesso de arranjo que pode ser estourado e consequentemente deve ser sanitizado ou evitado pelo desenvolvedor.

3. Resultados

A análise estática descrita na seção anterior foi implementada como um módulo para o compilador LLVM e executada sobre o conjunto de *benchmarks SPEC CPU INT 2006*. A linha de base utilizada para comparação do resultado foi a solução proposta por Mafra [Maffra et al. 2013], que não leva em consideração as dependências de controle e portanto não pode ser considerada consistente. A Figura 2 mostra o número de vértices contaminados em cada *benchmark*. Um vértice está contaminado se ele faz parte de um caminho que leva um vértice do conjunto G_e à um vértice do conjunto G_s. Claramente o número de vértices em caminhos vulneráveis encontrados pela análise de linha de base é significativamente inferior ao número encontrado pela análise consistente proposta neste trabalho. Isso se justifica porque a linha de base não considera os fluxos implícitos de informação determinados pelas dependências de controle. Além disso, a análise consistente revelou que do total de 3,559,715 vértices, 1,475,574 estão inseridos em caminhos vulneráveis. Isto é, 41% do código fonte da coleção de *benchmarks*.

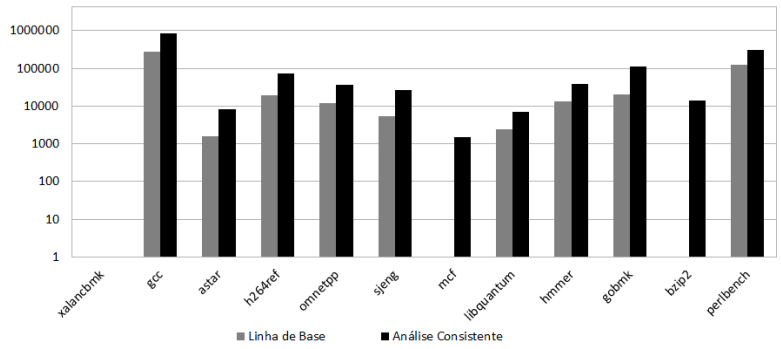


Figura 2. Comparação entre o número de vértices vulneráveis da solução de linha de base e da análise estática consistente.

4. Conclusão e Trabalhos Futuros

Está claro que desconsiderar os fluxos implícitos em qualquer análise de fluxo de informação é um erro que pode custar a consistência da análise. Este trabalho mostrou que isso é particularmente verdade para o problema de estouro de arranjos. Como trabalho futuro pretende-se criar um instrumentador de código. Assim, as aplicações serão analisadas e aqueles traços de instruções em caminhos detectados como vulneráveis, serão instrumentados com novas instruções capazes de impedir um estouro de arranjo. Além disso, busca-se utilizar a análise estática aqui proposta, na solução de outros importantes problemas de segurança computacional, tais como o vazamento de informação por canais laterais em sistemas criptográficos [Kocher 1996] .

Referências

Ferrante, J., Ottenstein, K. J., and Warren, J. D. (1987). The program dependence graph and its use in optimization. *TOPLAS*, 9(3):319–349.

Kocher, P. C. (1996). Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 104–113, London, UK, UK. Springer-Verlag.

Lattner, C. and Adve, V. S. (2004). LLVM: A compilation framework for lifelong program analysis & transformation. In *CGO*, pages 75–88. IEEE.

Maffra, I. K. T., Pereira, F. M. Q., and Oliveira, L. B. (2013). Detecção automática de vulnerabilidades em código protegido por canários. In *SBSeg*, pages 184 –197.

Quadros, G. S., Souza, R. M., and Pereira, F. M. Q. (2012). Dynamic detection of address leaks. In *SBSeg*, pages 61–75.

Silva, B. R. (2014). Um algoritmo linear para a construção de program slices. In *Anais do XVIII Simpósio Brasileiro Linguagens de Programação*.

Silva, Bruno Rodrigues, P. F. M. Q. O. L. B. (2013). Uma representação intermendiária para a detecção de vazamentos implícitos de informação. In *Anais do XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 212–225.

Um Mecanismo Simples e Eficiente para a Autenticação de Dispositivos na Comunicação por Campo de Proximidade

Silvio E. Quincozes[1] **e Juliano F. Kazienko**[1]

[1]Curso de Ciência da Computação – Universidade Federal do Pampa (UNIPAMPA)
Prédio A1 – CEP 97.546-550 – Alegrete – RS – Brasil

silvioereno@alunos.unipampa.edu.br, kazienko@unipampa.edu.br

Abstract. *Near Field Communication is a recent technology that uses radio waves at high frequency for data communication. One of the major research challenges in this area is the end point identification. In this sense, a hidden device within the legitimate devices coverage area makes possible to capture data as well as the device impersonation. This work presents a mechanism for devices authentication, aggregating relevant security properties for data exchange. The proposal is evaluated through a prototype. Low number of messages exchanged, low mechanism execution time and improvements related to other works reveal that the proposed mechanism is promising.*

Resumo. *A comunicação por campo de proximidade é uma tecnologia recente que utiliza ondas de rádio de alta frequência para a comunicação de dados. Um dos principais desafios de pesquisa nessa área consiste na identificação do ponto final. A presença de um dispositivo camuflado dentro da área de cobertura de dispositivos legítimos torna possível a captura das informações, bem como a personificação dos mesmos. Este trabalho apresenta um mecanismo que autentica os dispositivos, agregando propriedades de segurança importantes para a troca de dados. A proposta é validada através de um protótipo. O baixo número de mensagens trocadas, baixo tempo de execução do mecanismo e diferenciais em relação a outros trabalhos revelam que o mecanismo é promissor.*

1. Introdução

A tecnologia de Comunicação por Campo de Proximidade, do inglês, *Near Field Communication* (NFC), possibilita a troca de mensagens entre dispositivos, como celulares, *notebooks*, crachás, etc. Tal tecnologia de comunicação sem fio utiliza ondas de rádio de alta frequência, tradicionalmente 13,56 MHz, com um alcance máximo de dez centímetros. O NFC tem aplicações em diversos campos, como saúde, transporte e pagamentos eletrônicos. O NFC tem maior usabilidade e menor tempo de configuração, comparado a tecnologia *Bluetooth*, por exemplo [Eun et al. 2013][Coskun et al. 2013].

Um dos grandes desafios de pesquisa na área consiste no estabelecimento de segurança nas comunicações. A curta distância exigida para a troca de dados pode ser considerada uma vantagem. Desse modo, um atacante precisa estar muito próximo dos dispositivos legítimos para interceptar as mensagens trocadas, o que facilita sua detecção. No entanto, a presença de um dispositivo malicioso camuflado dentro da área de cobertura dos dispositivos legítimos pode viabilizar ataques, bem como a personificação dos dispositivos. Desta forma, o emprego de mecanismos destinados à garantia da autenticidade dos dispositivos se faz essencial.

Segundo [Miorandi et al. 2012], a comunicação segura através de radiofrequência requer soluções eficientes e de baixo custo computacional. Em muitos cenários que envolvem o uso de NFC, o consumo energético deve ser considerado, especialmente ao se utilizar dispositivos móveis. Atualmente, pontos que requerem atenção são: privacidade, autenticação de entidades, prevenção da espionagem e baixo custo energético [Alzahrani et al. 2013]. O presente trabalho tem por objetivo propor um mecanismo leve e eficiente que autentica mutuamente dispositivos NFC, agregando propriedades de segurança importantes para a troca de dados. De forma a aplicar o mecanismo proposto, foi implementado um protótipo. Nas demais seções tal mecanismo é introduzido.

2. Trabalhos Relacionados

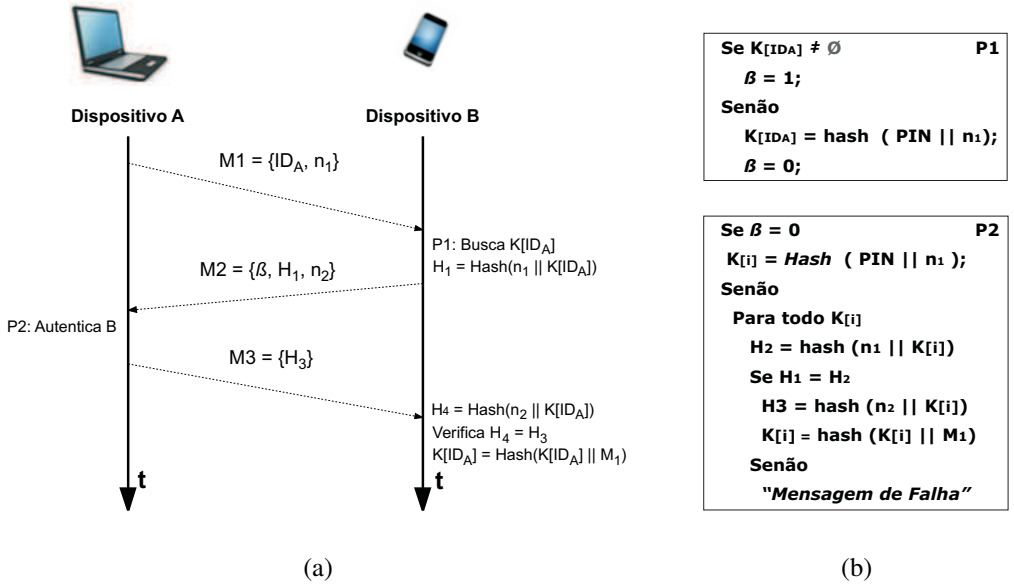
Em [Chen et al. 2010], é proposto um mecanismo para estabelecer a autenticação de dispositivos NFC e permitir transações financeiras. Foram aproveitadas as primitivas criptográficas da tecnologia GSM, utilizando o cartão *Subscriber Identity Module* (SIM) de celulares para identificá-los. Nesse trabalho, as chaves são geradas dinamicamente a cada autenticação. Como a operadora de telefonia participa na geração da chave compartilhada, o uso do mecanismo fica limitado a dispositivos da mesma operadora.

O trabalho de [Eun et al. 2013] propõe um método de privacidade condicional para proteger a privacidade do usuário usando pseudônimos. Essa proposta exige um terceiro confiável para a solicitação de um conjunto de pseudônimos. Com essa renovação o problema de rastreabilidade é controlado. Porém, essa abordagem requer aparelhos equipados com *Secure Element* (SE), o que inviabiliza o uso seguro dos demais dispositivos, como etiquetas NFC, por exemplo. Além disso, existe um custo computacional adicional para a solicitação de novos pseudônimos. Segundo os autores, um conjunto de 1000 pseudônimos exigira um espaço de 146,484 Kbytes em memória.

3. O Mecanismo Proposto

Suponha que dois dispositivos *A* e *B* necessitam se comunicar de forma segura através do modo de operação *Peer-To-Peer* [Coskun et al. 2013]. O dispositivo *A* possui identidade pública ID_A, como um terminal para compra de passagens, por exemplo. O dispositivo *B* é um aparelho pessoal, como um telefone celular, por exemplo, conforme ilustrado na Figura 1(a). Logo *B* deve ter sua identidade ID_B preservada. Nesse caso, *A* possui uma lista de chaves $L_A \leftarrow \{K_{[i]}, K_{[i+1]}...\}$. Cada chave representa um dispositivo conhecido. O dispositivo *B* possui outra lista de chaves $L_B \leftarrow \{K_{[ID_A]}, K_{[ID_C]}...\}$, onde cada registro possui referência ao ID de um dispositivo conhecido. Dessa forma, quando *A* envia ID_A, *B* deve procurar por $K_{[ID_A]}$ como chave para a autenticação. Na primeira comunicação $K_{[ID_A]}$ não existirá. Nesse caso, *B* atribui para o *bit* β de associação: $\beta \leftarrow 0$. Assim, ao receber a mensagem, *A* sabe que é preciso que seja definida e compartilhada uma nova chave. Para tal, a sugestão aqui proposta é o uso de um Número de Identificação Pessoal (PIN). O PIN deve ser gerado aleatoriamente e exibido na tela do dispositivo *B*, conforme *P1* na Figura 1(b). Em seguida, tal PIN é mostrado ao operador do dispositivo *A*, que atualiza o sistema para o cálculo de *K*, conforme *P2* na Figura 1(b).

Inicialmente, ao detectar a presença de um dispositivo próximo, *A* envia uma mensagem $M_1 \leftarrow \{ID_A, n_1\}$ em texto plano, onde n_1 é um *nonce*. O dispositivo *B* faz uma busca por $K_{[ID_A]}$ (*P1*, Figura 1(b)). Se $K_{[ID_A]} \neq \emptyset$, então $\beta \leftarrow 1$. Senão,



(a) (b)

Figura 1. A Figura (a) ilustra as mensagens trocadas. A Figura (b) detalha os algoritmos utilizados no mecanismo proposto.

$\beta \leftarrow 0$ e um PIN é gerado para definição de $K_{[ID_A]} \leftarrow Hash(PIN||n_1)$. É calculado $H_1 \leftarrow Hash(n_1||K_{[ID_A]})$ e gerado outro *nonce* n_2. Em seguida *B* emite $M2 \leftarrow \{\beta, H_1, n_2\}$. Quando $\beta = 0$ (P2, Figura 1(b)), insere-se $K_{[i]} \leftarrow Hash(PIN||n_1)$ na lista. Para tal, é necessário que o usuário digite em *A*, o PIN gerado e exibido por *B*. Quando $\beta = 1$, o dispositivo *A* computa $H_2 \leftarrow Hash(n_1||K_{[i]})$ e verifica se $H_1 = H_2$. Isso se repete para todo índice *i* dos registros da lista L_A até que seja satisfeita a igualdade, onde *B* é considerado autêntico. Se a igualdade não for satisfeita, ocorre então uma falha na autenticação de *B* e o processo é interrompido. Por fim, *A* calcula $H_3 \leftarrow Hash(n_2||K_{[i]})$, atualiza $K_{[i]} \leftarrow Hash(K_{[i]}||M_1)$ e envia $M_3 \leftarrow \{H_3\}$. O dispositivo *B* computa $H_4 \leftarrow Hash(n_2||K_{[ID_A]})$ e verifica se $H_3 = H_4$. Se a igualdade for satisfeita, *A* é autenticado e ocorre a atualização de $K_{[ID_A]} \leftarrow Hash(K_{[ID_A]}||M_1)$.

4. Resultados Preliminares e Trabalhos Futuros

A fim de validar o mecanismo da Figura 1(a), um protótipo foi implementado. Para tal, utilizou-se um telefone celular da marca Sony modelo Xperia M, com o sistema operacional Android 4.2 instalado, e um notebook da marca Sony Vaio, modelo *svf15213cbw*, com sistema operacional Windows 8. O estabelecimento da comunicação entre tais dispositivos está calcada no modo de operação entre dispositivos NFC denominado *Peer-to-Peer* [Coskun et al. 2013]. A programação do protótipo se deu nas linguagens Java e C Sharp, respectivamente. Para computar os resumos foi utilizado o *Message-Digest algorithm 5* (MD5), entretanto qualquer algoritmo *hash* poderia ser utilizado. O tamanho total do executável instalado é de 1,25 Mbytes no celular e 25 Kbytes no notebook. O tempo médio total para a execução do mecanismo proposto é de 253 ms.

Visto que a descoberta de *K* implica no comprometimento da autenticidade de quaisquer dos dispositivos, o mecanismo proposto efetua a renovação dinâmica de chaves a cada autenticação. Dessa forma, mesmo que um atacante descubra *K* anterior, ela não

servirá para a próxima autenticação. Tal processo é independente de terceiro confiável. Além disso, o usuário do celular consegue provar sua autenticidade sem expor sua identidade, mantendo assim a privacidade e evitando o rastreamento. Devido ao baixo número de mensagens trocadas, o mecanismo se torna eficiente do ponto de vista do consumo energético. A Tabela 1 compara trabalhos existentes com o mecanismo proposto aqui.

Tabela 1. Propriedades dos Mecanismos.

	[Eun et al. 2013]	[Chen et al. 2010]	**Mecanismo Proposto**
Privacidade do Usuário	Em Risco	Parcial	Satisfatória
Terceiro Confiável	Dependente	Dependente	Independente
Simplicidade	Complexo	Médio	Simples
Espaço de Armazenamento	Demasiado	Pouco	Pouco
Autenticação Mútua	Possui	Possui	Possui
Dependência de SE	Dependente	Dependente	Independente
Renovação de Chaves	Razoável	Razoável	Aceitável

Observa-se que a renovacão de chaves é um ponto que requer melhorias. O uso de um terceiro confiável é uma alternativa, entretanto deve ser considerada a possibilidade da indisponibilidade do mesmo. O mecanismo proposto independe de terceiro confiável. Porém, se um atacante descobrir K e escutar as mensagens trocadas no processo de autenticação, ele é capaz de computar o novo K. Uma solução para tal problema consiste no uso de um PIN gerado em um dos dispositivos e digitado pelo usuário no outro dispositivo. É importante destacar que a geração do PIN deve acontecer sempre que houver autenticação, o que pode ser inconveniente do ponto de vista do usuário.

Como trabalhos futuros, pretende-se aplicar o mecanismo em um ambiente hospitalar. Desse modo, um médico poderia de forma rápida e segura recuperar dados de pacientes como seus prontuários médicos através de dispositivos móveis. Adicionalmente, deseja-se utilizar etiquetas NFC junto aos leitos dos pacientes, adequando o mecanismo proposto a tais dispositivos. Nesse cenário, a autenticação, privacidade, integridade e confidencialidade são propriedades que devem ser garantidas [Alzahrani et al. 2013].

Referências

Alzahrani, A., Alqhtani, A., Elmiligi, H., Gebali, F., and Yasein, M. S. (2013). NFC security analysis and vulnerabilities in healthcare applications. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 302–305. IEEE.

Chen, W., Hancke, G., Mayes, K., Lien, Y., and Chiu, J.-H. (2010). NFC mobile transactions and authentication based on GSM network. In *Second IEEE International Workshop on Near Field Communication (NFC)*, pages 83–89. IEEE.

Coskun, V., Ozdenizci, B., and Ok, K. (2013). A Survey on Near Field Communication NFC Technology. *Wireless Personal Communications*, 71:2259–2294.

Eun, H., Lee, H., and Oh, H. (2013). Conditional privacy preserving security protocol for NFC applications. *IEEE Transactions on Consumer Electronics*, 59(1):153–160.

Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516.

Relação custo/benefício de técnicas utilizadas para prover privacidade em computação nas nuvens

Vitor Hugo Galhardo Moia[1], **Marco Aurélio Amaral Henriques**[1]

[1]Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (Unicamp)
Campinas, SP, Brasil 13083–852

{vhgmoia,marco}@dca.fee.unicamp.br

Resumo. *Com a crescente utilização da computação nas nuvens, devido a suas atraentes características, surge uma grande preocupação com a segurança e privacidade nesse meio. Possíveis ataques realizados pelos próprios provedores de serviço ou por terceiros tornam os usuários relutantes na utilização desta tecnologia. Em meio a tais problemas, este trabalho apresenta uma discussão sobre os principais problemas e soluções relativos à privacidade nas nuvens e faz uma comparação preliminar destes com base em estimativas dos custos e do grau de privacidade de cada um.*

1. Introdução

A computação nas nuvens é uma boa opção para o armazenamento de dados de forma escalável e dinâmica. Com isso, evitam-se custos com infraestrutura e manutenção locais, além de ser possível pagar conforme a utilização. Outra vantagem é a disponibilidade dos dados a partir de qualquer ponto na internet. Contudo, um fator que impede uma maior adoção dessa tecnologia é a preocupação quanto à segurança e privacidade. Um potencial acesso a informações sensíveis pelos provedores é um grande problema, pois estes têm fácil acesso aos dados. Além disso, há a possibilidade de falhas na segurança da infraestrutura que permitam a invasores se apoderar de dados armazenados.

Com o objetivo de minimizar riscos na segurança foram propostas na literatura várias técnicas. Neste trabalho são discutidos os principais problemas relativos ao sigilo de dados na nuvem e alguns métodos para resolvê-los. Em seguida são feitas estimativas dos custos e benefícios de cada método a fim de permitir uma comparação preliminar entre os mesmos.

2. Problemas relacionados a privacidade na nuvem

Geralmente, os usuários carregam seus dados na nuvem de forma clara e confiam em seus provedores de serviço quanto à segurança de suas informações. Para não precisar desse tipo de confiança e aumentar a proteção dos dados armazenados, foram propostas na literatura várias formas para se aumentar o sigilo dos mesmos. Nesta seção, será realizada uma discussão sobre os principais problemas de privacidade na nuvem, apresentando algumas técnicas utilizadas para saná-los.

2.1. Sigilo do conteúdo

Consiste no acesso controlado aos dados armazenados na nuvem, os quais só podem ser acessíveis ao(s) seu(s) dono(s). Havendo este controle, os usuários não terão que confiar

cegamente em seus provedores e suas informações ficarão mais protegidas contra atacantes externos. Várias técnicas foram desenvolvidas para essa solução, sendo as duas principais a criptografia e o armazenamento distribuído dos dados particionados em larga escala (fragmentação). A primeira garante o sigilo dos dados por meio de codificação baseada em um segredo, porém traz alguns problemas como o menor desempenho em geral (há tempos extras para cifrar e decifrar) e a dificuldade de compartilhar, buscar e indexar dados cifrados. Há várias maneiras de se utilizá-la, sendo a mais comum cifrar um dado antes de enviá-lo para a nuvem e decifrá-lo apenas após o usuário tê-lo trazido de volta para seu sistema local [Padmaja and Koduru 2013]. Já a fragmentação dos dados pode utilizar técnicas como *erasure code* e *secret sharing* para prover alguma redundância entre diversos provedores de armazenamento. Assim, para recuperar o dado original é necessário buscar apenas parte desses fragmentos, acarretando na não dependência de um determinado provedor de armazenamento e em uma maior disponibilidade dos dados [Abu-Libdeh et al. 2010]. Também é possível combinar estas duas técnicas, mas permanecendo as dificuldades de compartilhar, indexar e buscar dados armazenados na nuvem [Schnjakin et al. 2011].

2.2. Sigilo do nome dos dados

Um problema pode ocorrer caso o nome do dado ou sua extensão contenha alguma informação relativa ao seu conteúdo. Para minimizar tal problema, o nome e a extensão do arquivo devem ser alterados, tomando cuidado ao renomear arquivos fragmentados, pois estes podem ser renomeados contendo nomes como dados-parte1, dados-parte2 etc., revelando não só o nome do arquivo, mas também que foi dividido. Uma das possíveis soluções para resolver esse problema é através de técnicas criptográficas. Assim, o nome do arquivo seria cifrado com uma chave secreta e isso evitaria que o nome revelasse algo sobre o arquivo e até mesmo o problema de se saber que o arquivo foi particionado. A chave secreta pode ou não ser a mesma usada na criptografia dos dados, dependendo dos procedimentos de gerência de chaves adotados.

2.3. Sigilo de localização durante acesso aos dados

Outro problema diz respeito à proteção do usuário quanto à revelação de sua localização durante o acesso aos seus dados. Há vários recursos para esse fim, como o Tor e o VPN-Proxy. Essas tecnologias têm como finalidade proteger o usuário contra vigilância e ataques. O TOR realiza o roteamento de uma mensagem através de várias máquinas aleatórias e utiliza várias camadas de encriptação para proteger o usuário e sua comunicação [Murdoch and Danezis 2005]. Já o VPN combinado com proxy cria um canal de comunicação criptografado entre dois pontos: o computador do usuário e um serviço de VPN-Proxy (como o VPNBook.com) que oculta do provedor da nuvem a localização do usuário [Duffield et al. 2005]. A utilização dessas duas tecnologias, ou outras que possuam características semelhantes, é necessária para a minimização deste problema de privacidade e obtenção do anonimato.

2.4. Sigilo sobre a posse de um dado

Se houver facilidade de um provedor ligar os dados armazenados na nuvem a seus respectivos donos, teremos outro problema de falta de privacidade. Uma das maneiras de se resolver este problema é com a adoção de identidades federadas, onde se tem provedores

de identidade separados de provedores de serviços. Os provedores de identidade, além de autenticar o usuário, têm a função de guardar os atributos dos usuários e repassá-los para os provedores de serviços de forma controlada. Há situações em que o provedor de identidade não precisa repassar nenhum atributo do usuário que possibilite ao provedor de serviços descobrir a identidade verdadeira do usuário. Além disso, para situações em que se quer uma maior privacidade, o provedor de identidade pode ser configurado para não armazenar informações sobre os acessos de seus usuários.

3. Soluções para a privacidade

Nesta seção são discutidas e comparadas técnicas para se prover algum tipo de privacidade na nuvem. Para cada uma delas é estabelecido pelos autores um Custo Relativo (CR) de acordo com uma estimativa subjetiva dos custos de tempo e espaço em relação às outras (Tabela 1). É considerado que a infraestrutura necessária para cada uma já esteja instalada e pronta para ser utilizada não sendo necessário contabilizar seu custo. Também é estabelecido um grau subjetivo de Privacidade Relativa (PR) de acordo com o nível estimado de privacidade obtido com cada técnica em relação às outras. A relação custo/benefício (R) é obtida pela divisão de CR por PR.

Tabela 1. Custo e privacidade relativos das técnicas utilizadas para prover privacidade na nuvem

Sigla	Técnica	CR	PR
T_5	Criptografar conteúdo	04	3,5
T_4	Fragmentar com redundância	10	2,5
T_3	Ocultar acesso via TOR	20	1,5
T_2	Ocultar acesso via VPN-Proxy	08	1,0
T_1	Ocultar posse dos dados (identidade federada)	02	1,5
T_0	Ocultar nome dos dados (criptografia de nomes)	01	1,0

Algumas técnicas como T_2, T_3, T_4 e T_5 dependem do tamanho do arquivo para se calcular o tempo e espaço requeridos e, por essa razão, foi considerado um arquivo de mesmo tamanho para a estimativa do peso de todas elas. A técnica de ocultar acesso está sendo considerada de duas maneiras, de acordo com a tecnologia utilizada.

Para a identificação das possíveis abordagens, é proposta uma forma para se criar nomes baseados nas combinações das técnicas. Há 48 possibilidades incluindo o uso de nenhuma técnica. Para ocultar a origem do acesso, as duas formas, TOR e VPN-Proxy, não podem estar na mesma combinação, e cada combinação usa um modelo de código binário como nome, sendo que 0 indica a não utilização de determinada técnica e 1 indica a sua utilização. Cada técnica ocupa uma posição em uma palavra de 6 bits, cuja ordem é T_5, T_4, T_3, T_2, T_1, T_0 e o valor decimal resultante é a representação da combinação. Considera-se que a PR_i e a CR_i de uma combinação i de técnicas são dadas pelas somas das respectivas PR e CR de cada técnica.

A Fig. 1 mostra todas as soluções possíveis com seus respectivos custos e privacidades relativas ($CR_i \times PR_i$), e também apresenta o valor de $R_i = CR_i/PR_i$ de cada combinação i na forma do tamanho dos círculos na figura. Nota-se que as combinações mais interessantes são as localizadas no quadrante inferior direito (os números indicam as combinações).

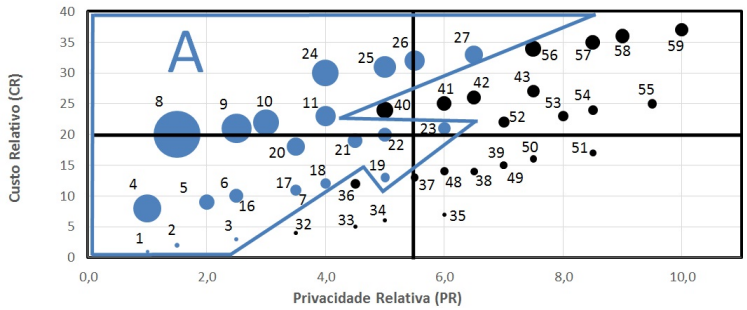


Figura 1. Custo Relativo e Privacidade Relativa para diferentes combinações

As técnicas T_0, T_1, T_3, T_4, e T_5 causam individualmente uma variação em R_i (i = 59) de +8%, +11%, -46%, -3% e +37%, respectivamente, quando são removidas desta configuração completa que usa todas as técnicas (exceto T_2). Logo, conclui-se que a adoção de técnicas como T_3 e T_4 deve ser feita com muito cuidado, pois os custos podem não compensar os benefícios. O mesmo padrão pode ser visto quando se usa T_2 no lugar de T_3, alertando para o alto custo/benefício destas duas técnicas em relação às demais.

A Fig. 1 também mostra o impacto da adoção ou não de T_5 em R. A Técnica T_5 está inativa na zona A e ativa fora dela (o bit relativo a T_5 é 0 nas combinações de 1 a 31). A inclusão de T_5 contribui significativamente para redução da razão R. Este tipo de comportamento deve ser considerado na escolha desta e de outras técnicas.

4. Conclusão

Neste trabalho foram discutidos alguns problemas relativos à privacidade nas nuvens e apresentadas algumas técnicas para amenizá-los. Uma análise dos níveis de privacidade e de seus custos relativos mostrou que algumas combinações de técnicas são bem mais eficazes em termos da relação custo/benefício que outras. Como trabalho futuro é preciso tentar obter custos absolutos mais precisos de forma a se ter uma visão mais completa que permita uma melhor tomada de decisão sobre que técnicas utilizar. Além disso, é preciso investigar formas eficientes de se combinar 2 ou mais das técnicas aqui descritas para se obter um maior grau de privacidade.

Referências

Abu-Libdeh, H., Princehouse, L., and Weatherspoon, H. (2010). Racs: A case for cloud storage diversity. In *Proc. of ACM*, SoCC '10, pages 229–240, New York, NY, USA. ACM.

Duffield, N., Greenberg, A., Goyal, P., Mishra, P., Ramakrishnan, K., and Van der Merwe, J. (2005). Virtual private network. US Patent 6,912,232.

Murdoch, S. and Danezis, G. (2005). Low-cost traffic analysis of tor. In *Security and Privacy, 2005 IEEE Symposium on*, pages 183–195.

Padmaja, N. and Koduru, P. (2013). Providing data security in cloud computing using public key cryptography. *IJESR*, 4(01).

Schnjakin, M., Alnemr, R., and Meinel, C. (2011). A security and high-availability layer for cloud storage. In *Proc. of*, WISS'10, pages 449–462, Berlin, Heidelberg. Springer-Verlag.

Decentralized management of One-Time Pad key material for a group

Jeroen van de Graaf

[1]Departamento de Ciência da Computação, UFMG

jvdg@dcc.ufmg.br

Abstract. *Suppose a group of users share copies of a large file of truly random bits, possibly distributed through portable USB sticks or external hard drives. In this note we present a randomized, distributed key management scheme allowing these users to use this file as a One-Time Pad key, without fear of two users using the same key material twice.*

1. Motivation and problem

Consider the following setting: a small group of people, who meet on some regular basis, collaborate on some project. They wish to protect the texts they themselves produce: emails, chats, some typed documents maybe. Suppose that these people are closely watched by three- and four-letter agencies with lots of expertise in cryptography, like journalists working on classified information. In this situation it makes perfect sense to use the One-Time Pad (OTP). The amount of information to be encrypted is very small relative to current storage media such as SD cards, USB memory sticks or external hard drives, the latter being able to store 1 terabyte. So key transportation is possible through personal meetings or through couriers, personal or commercial (FedEx).

As a concrete example, suppose that these persons share a 1 GB random file which has been reliably distributed to each member by copying it to USB sticks on an air-gapped, and otherwise protected, secure hardware platform. They would like to interchange documents over the internet, encrypting them using the file as an OTP key. We assume this OTP key to be secret towards outsiders, in particular the spy agency.

However, it is well-known that an OTP key should never be used twice, since in that case the OTP loses its security properties, and a statistical analysis allows partial recovery of the plaintexts[2]. So the problem they need to solve is this: How to avoid collision of the key material? How to avoid that different members use the same part of the OTP key to encrypt different documents? How to decide who uses which part of the OTP key file considering these constraints?

One solution is to have a central server controling key distribution, allocating parts of the OTP key to the members. However, this would require the group members to be online, and constitutes a single point of failure. It also constitutes a single point for a spy agency to monitor: even not having access to the OTP key file, they might be able to find out who communicates with whom, and the sizes of the messages sent. Traffic analysis, in other words. So we prefer a solution without a central server.

Another solution is letting a sender choose a start position p at random, and use the bits from that position onward as a OTP key. This p is included as meta data, sent in the clear (for simplicity of exposition–we can do better), to inform the recipient where to

find the position of the decrypt key. However, in this case there is a risk that two members choose the same start position, and confidentiality is compromised. For concreteness, suppose that the 1GB key file is divided into 1024 blocks of 1MB each, and that the group members send 1MB messages. From the birthday paradox (see for instance [3, 4] we know that after only $1.17\sqrt{1024} \approx 37$ message we have 50% chance that a least two messages collide and have been compromised. The question addressed in this note is: can we do better? The answer is YES.

2. Towards a better solution

In order to reduce the collision probability, we let a sender choose several start positions, and derive a OTP subkey from each. In other words, the sender chooses N start positions at random, where $N = 16$ (this choice will be discussed later). Again, the random start positions $p_1, \ldots, p_N$ are added as metadata to the encrypted message.

Now let M be a plaintext message of size 1MB, let K be the OTP key of 1GB, and let C be the ciphertext. We divide K in 1024 blocks of size 1MB each, and define K_p as the pth block, to be used as a OTP sub-key. Now randomly choose N different start positions $p_1, \ldots, p_N$ with $p_i \in \{1 \ldots 1024\}$. Then compute the net OTP key as $K^* := K_{p_1} \oplus K_{p_2} \oplus \ldots \oplus K_{p_n}$ and the ciphertext $C := M \oplus K^*$, where $\oplus$ denotes the bitwise xor operation.

Observe that now, for two parties to collide, they would have to choose the *same* subset of N positions in a total of 1024 possibilities. So the space of possibilities is of size $\binom{1024}{N}$, and applying the approximation for the birthday paradox to this example, we obtain $1.17 \cdot \binom{1024}{N}^{1/2}$. For $N = 16$ this gives $1.17 \cdot \binom{1024}{16}^{1/2} \approx 1.17 \cdot (10^{34.79})^{1/2} \approx 2.91 \cdot 10^{17}$. This means that around $2.91 \cdot 10^{17}$ messages would be needed to have a 50% change of a collision occuring. Even though 50% is much higher than desirable, it shows that our approach is promising.

We also address the efficiency of the scheme, analysing what percentage of the available random key material can be used without making compromising security. We claim that, even for reasonably small values of N, the users can essentially use 99% of the random key material, while the probability of having collisions remains neglible.

3. More on the collision probability

The preceding approximation of the collision probability is incomplete, in the sense that it calculates how many message can be sent in order to have a collision with 50% chance. However, a collision is a catastrophic event, and its probability should be kept very low: $\varepsilon = 10^{-10}$ or less.

Let S be the total set size, and k be the number of elements chosen. (For the birthday paradox $S = 365, k = 23$). Let $\varepsilon = p(S, k)$ be the probability of having a collision. A well-known approximation for $\overline{p}(S, k) = 1 - \varepsilon$ based on Taylor series [3] is $\overline{p}(S, k) = e^{-(k(k-1))/2S}$. Approximating $k - 1 \approx k$ and taking logs on both sides, we get $k^2 = 2\ln(\frac{1}{\varepsilon}) \times S$. For the birthday paradox, with $\varepsilon = 1/2$, this gives us $k = \sqrt{2\ln(2)}\sqrt{S} \approx 1.17\sqrt{S}$, the approximation used in the first example using only 1 position, and which is known to give the correct answer, $k = 23$, for $S = 365$.

Now using the numbers of the second example we have $S = \binom{1024}{16} \approx 10^{34}$, and setting $\varepsilon = 10^{-10}$ we obtain $k = \sqrt{2\ln(\frac{1}{1-\varepsilon})}\sqrt{S} = \sqrt{2(10^{-10})}\sqrt{10^{34}} = 1.41 \cdot 10^{12}$. Here we used that $\ln(\frac{1}{1-\varepsilon}) \approx \ln(1+\varepsilon) \approx \varepsilon$ for ε very small. This formula shows that the failure probability $\overline{q}$ and the the size of the possibility space S balance fairly: if we want to fix the number of messages k, but want to reduce ε by a factor 10 (say), we need to multiply S by 10. The value $k = 1.41 \cdot 10^{12}$ is higher than we need, as the discussion in the next section shows.

4. Higher order collisions

The preceding analysis cannot be the complete picture since the limit on the number of message seems to be very high, whereas we know for sure that after 1024 messages of 1MB we *must* have exhausted the 1GB key material. The point is that the analysis above only explores collisions between *two* different messages, whereas much more complicated collisions are conceivable.

Let $p_{r1}, \ldots, p_{rA}$ denote the OTP key positions chosen when encrypting message r, where A is the total number of positions (above $A = 1024$). Let us describe these N positions as a 0/1 row vector P, where $\overrightarrow{P}_{ri} = 1$ iff $i \in \{p_{r1}, \ldots, p_{rN}\}$. By vertically listing the row vectors, we obtain a $T \times A$ matrix over $\mathbb{F}_2$ called $\mathcal{P}$, after T messages have been sent. As soon as $\mathcal{P}_{T\times A}$ has a linear dependency we have a problem, since it means that there exists a non-empty subset $I \subseteq \{1 \ldots T\}$ such that $\bigoplus_{i\in I} C_i = \overrightarrow{0}$, implying that $\bigoplus_{i\in I} M_i = \bigoplus_{i\in I} K_i^*$. So if $\bigoplus_{i\in I} K_i^* = \mathbf{0}$ the adversary knows that $\bigoplus_{i\in I} M_i = \mathbf{0}$. The collision described in the previous section is a special case of this, with $\#I = 2$.

So we must answer the following question: if we have row vectors of size A, and we keep on adding rows, what is the probability that the resulting matrix has a linear dependency after having added T rows? Obviously, T cannot exceed A but the question is how close we can get.

For *random matrices*, where each entry is 0 or 1 with 50% chance, the formula for the expected dimension (rank) of $\mathcal{P}_{T\times A}$ is known ([1], §3.5). However, a downside using this approach is that each row has an expected Hamming weight of 512, so we need to do $N = 512$ file seeks and xors on average. We would like to reduce N to a much smaller value, though very small values, like $N = 1, 2$ or 3, lead to collisions. But how about $N = 8$ or $N = 12$? How large can T be before we have a linear dependency?

Though similar problems have been studied in the context of low-density parity check codes, we have not yet been able to find a relevant reference or derive an exact formula. Instead we did some computer simulations: we randomly generate a new row vector of Hamming weight N, we check if the new vector is independent and use straightforward Gaussian elimination over $\mathbb{F}_2$ to eliminate all zeroes below the diagonal. The results of these simulations for $A = 1024$ over 100 experiments are as follows:

N	1	2	3	4	5	6	7	8	10	12
avg	39.6	459.2	937.4	996.8	1014.9	1018.2	1021.4	1021.1	1021.3	1021.7
min	3	110	906	984	1008	1010	1015	1013	1014	1015
max	110	639	961	1013	1023	1023	1024	1023	1023	1023

These results show that for $A = 1024$ and a value for N as low as 8 the dimensionality of the matrix essentially behaves as a random matrix, meaning that the probability of

linear dependencies is extremely low until T approaches A. Given the data it seems safe to conjecture that for $N \geq 8$ and $T = 1000$, the probability that a dependency occurs is negligible. Finding an analytical formula for this probability is one of our main research priorities; otherwise more efficient and more simulations are needed.

5. Some observations and conclusion

(1) The random OTP file should be stored in encrypted form, so that if it falls into the wrong hands not everything is compromised. Security then degrades to cryptographic security. Using counter mode with a strong symmetric cipher is a good option because we want random access to the file. (2) The positions used could be included using symmetric encryption to make the adversary's task harder. (3) One should include message authentication such as an adaptation of Galois Counter Mode, where the OTP blocks are interpreted as if they were the blocks generated by the counter mode. (4) To protect against leakage of individual bits one could apply unconditional all-or-nothing transforms [5] before encryption. (5) Choosing the right block size (1MB in the example) depends on the application. Maybe more than one size is useful.

Bruce Schneier once discarded the OTP as follows: "What a one-time pad system does is take a difficult message security problem – that's why you need encryption in the first place – and turn it into a just-as-difficult key distribution problem. It's a 'solution' that doesn't scale well, doesn't lend itself to mass-market distribution, is singularly ill-suited to computer networks, and just plain doesn't work." I see his point, but there are situations in which using the OTP makes perfect sense. People use email and WhatsApp often with people they meet on a frequent basis. Why can't they use the OTP here? We should work harder to make this a viable option.

Note added in proof: One of the referees pointed out that the analysis presented here does not include so-called *meet-in-the-middle attacks*, as presented in [6]. These attacks imply that the actual security level of this scheme is much lower than what is claimed, so assessing their impact is currently a top research priority.

References

[1] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. NIST Special Publication 800-22 Revision 1a, April 2010.

[2] C. Shannon, *Communication Theory of Secrecy Systems*. Bell System Technical Journal 28 (4): 656–715..

[3] P. Halmos. *I Want to Be a Mathematician.* Springer-Verlag. ISBN 978-0387960784.

[4] Anonymous. *Birthday Problem* `http://en.wikipedia.org/wiki/Birthday_problem`

[5] Stinson, D. *Something About All or Nothing (Transforms)*. Designs, Codes and Cryptography, Volume 22 Issue 2, March 2001, Pages 133–138

[6] K. Nishimura and M. Sibuya, *Probability To Meet in the Middle*. J. Cryptology, vol 2 (1), 1990.

Software implementation of SHA-3 family using AVX2

Roberto Cabral, Julio López *

[1]Institute of Computing, University of Campinas
cabral@lasca.ic.unicamp.br, jlopez@ic.unicamp.br

Abstract. *The Keccak algorithm was the winner of the competition organized by NIST to choose the new standard hash algorithm, called SHA-3. In this work, we present the details of our software implementation in conformity with draft FIPS 202. We follow two approaches for the implementation of SHA-3, the first one computes the digest for a single message, and the other one computes in parallel four digests from four different messages. The performance for the single implementation was accelerated using vector instructions of 128/256 bits, and it is as fast as the best implementation optimized for 64 bits published on eBASH. The parallel implementation is about* $2.5\times$ *faster than the single message implementation. The cryptographic primitive extendable-output functions, which is part of the draft FIPS 202, were also implemented.*

1. Introduction

The family of hash functions SHA (Standard Hash Algorithm) [FIPS 2008], was standardized by the NIST (National Institute of Standards and Technology) and currently is used in many applications and protocols. Recently, several attacks on hash algorithms of SHA family were found. In 2005, [Biham et al. 2005] and [Rijmen and Oswald 2005] showed collision attacks of reduced versions of SHA-1. In the same year, [Wang et al. 2005] showed an attack that theoretically breaks the resistance to collision. The second version of SHA, SHA-2, is based on SHA-1 and already had attacks in its reduced versions, as is shown in [Indesteege et al. 2009]. In 2007, NIST started a new competition to select the new version of SHA algorithm, called SHA-3 [NIST 2007]. After two rounds of competition, five finalists were chosen: BLAKE, Grøstl , JH, Keccak and Skein. In 2012, Keccak [Bertoni et al. 2008] was announced as the winner.

This work shows how to take advantage of the new vector instructions (AVX/AVX2) introduced on Intel® Architecture Processors to implement the SHA-3 family. We developed a sequential and a parallel versions of the SHA-3 hash function for the four security levels 112, 128, 192 and 256 bits; in addition, the extendable-output functions (XOFs) were implemented for 128 and 256 bit security levels.

2. AVX2

In 2013 was released the newest Intel micro-architecture, called Haswell. This architecture contains the AVX2 (Advanced Vector Extensions 2) instruction set, which operates on 128-bit or 256-bit registers. Unlike the former AVX, AVX2 has vector instructions to perform integer arithmetic operations and permutations of words (8 - 64 bits) within registers. Using such instructions allow us to implement SHA-3 exploiting the data level parallelism present; in Table 1 one can see the instructions used in our implementation.

*The authors were supported in part by the Intel Labs University Research Office.

Category	Instructions	Latency
Logical	`XOR`, `AND`, `ANDN`	1,1,1
Shift	`SHIFT`, `VSHIFT`	1,2
Permutation	`SHUFFLE`,`VPERM`	1,3
Merge	`UNPACK`, `VBLEND`, `PRBLEND`	1,1,3

Table 1. The main AVX2 instructions used in our implementation of SHA-3.

3. SHA-3

According to the draft FIPS 202 [FIPS 2014], SHA-3 family consists of six functions, four of them are hash functions and the others are extendable-output functions. The hash functions are SHA3-224, SHA3-256, SHA3-384 and SHA3-512 and the XOFs are SHAKE128 and SHAKE256, in Table 2 are shown the parameters of these functions.

An extendable-output function maps an arbitrary-length message producing a variable-length digest. This function can be used when an application requires a cryptographic hash function with a non-standard digest length. We note that the security of these special functions is directly related to the size of the digest.

The SHA-3 family shares a sponge construction structure, which is a simple iterated construction for building a function with variable-length input and arbitrary-length output based on a permutation f operating on a state of $r + c = 1,600$ bits [Bertoni et al. 2007]; the state can be visually represented as a 5×5 matrix of 64-bit words.

The permutation function is divided into five steps: θ, ρ, π, χ and ι; the following, a short description of these steps:

1. In the θ step is computed an XOR of each word of the state with the parity of the left column and the right column rotated one bit.
2. In the ρ step each word of the state is rotated a fixed amount of bits.
3. In the π step the words of the state are permuted.
4. In the χ step is processed a non-linear function between the elements of the same row.
5. In the ι step is computed an XOR between the first element of the state with a constant value.

4. Implementations

The SHA-3 algorithm processes a state of 25 words of 64 bits using a permutation function f, which is composed of some steps that can be vectorized. We stored the state among

Function	Bitrate (r)	Capacity (c)	Security Level
SHA3-224	1,152	448	112
SHA3-256	1,088	512	128
SHA3-384	832	768	192
SHA3-512	576	1,024	256
SHAKE128	1,344	256	$\min(N/2, 128)$
SHAKE256	1,088	512	$\min(N/2, 256)$

Table 2. SHA-3 parameters.

registers in different ways for each implementation.

4.1. Single message hash computation

We developed two implementations of hash function with a single message; in the first one we represent the state as 13 registers of 128 bits, this allow us to vectorize the steps θ, ρ and χ processing two words per instruction; in the other one, the state is represented as 7 registers of 256 bits, thus we can process four words per instruction. The step π can not be vectorized, but there is an AVX2 instruction to perform this permutation, however, this instructions is too expensive for registers of 256 bits, as we shown in Table 1.

These two implementations can also be adapted to the XOF functions, the only difference is that in the squeezing phase the computation is performed just on the last state produced after the absorbing phase, thus the computation is mainly performed in registers.

4.2. 4-way hash computation

In this implementation, the state is represented as 25 registers of 256 bits and the operations between registers are performed totally in parallel, achieving the computation of four digests. Here the π permutation is implemented faster than in the single message implementation, additionally we can compute four words per instruction in all the steps, thus giving a significant speedup; the only drawback of this implementation is the amount of registers needed to store the four states, because the Haswell micro-architecture has only 16 available registers.

5. Preliminary results

We benchmark our implementations on a Core-i7 4770 processor, following the guidelines on [Bernstein and Lange 2014]. In Figure 1 we show the cycles per byte to compute the digests from messages of size 4KB to 2GB.

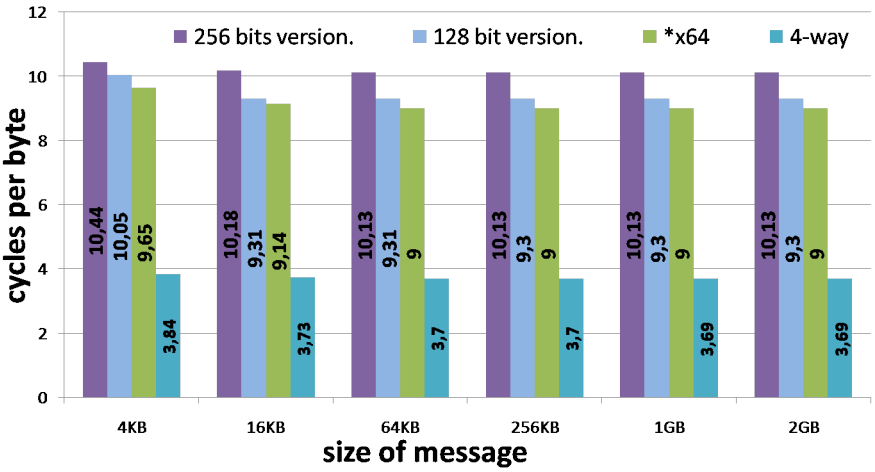


Figure 1. Cycles per byte to compute the digest from messages of size 4KB to 2GB.

The x64 implementation, developed by Ronny Van Keeris, is the fastest on eBASH for this processor. This implementation was optimized for 64 bits and does not use vector instructions.

	Absorbing	Squeezing
SHAKE128	7.79	7.7
SHAKE256	9.62	9.24

Table 3. Cycles per byte of SHAKE from a message of 4 KB.

Table 3 shows the cycles per byte to compute the absorbing and the squeezing phase in the XOFs implementations for a message of 4KB; as one can see the absorbing phase is more expensive than the squeezing phase, this happens because in the squeezing phase we do not need to process the message with the state, we only get the first r bits from state and then, process the state again through the function f until get all the digest required.

6. Conclusion

These preliminary results show that the use of vector instructions are useful for the efficient implementation of SHA-3. Using the AVX/AVX2 instructions allow us to achieve almost the same performance than the fastest 64-bit implementation for a single message setting. We observed that unlike the 64-bit implementation, the use of permutation instructions between vector registers is an expensive operation, since each permutation takes 3 clock cycles. For the 4-way setting, we obtained $2.5\times$ of speedup against the fastest single message implementation. This work is currently in progress and we are still looking for new optimization techniques to improve the results.

References

Bernstein, D. J. and Lange, T. (2014). ebacs: Ecrypt benchmarking of cryptographic systems.

Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G. (2007). Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer.

Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G. (2008). Keccak specifications. *Submission to NIST*, 42.

Biham, E., Chen, R., Joux, A., Carribault, P., Lemuet, C., and Jalby, W. (2005). Collisions of sha-0 and reduced sha-1. In *Advances in Cryptology–EUROCRYPT 2005*, pages 36–57. Springer.

FIPS, P. (2008). 180-3. *Secure Hash Standard*.

FIPS, P. (2014). 202. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*.

Indesteege, S., Mendel, F., Preneel, B., and Rechberger, C. (2009). Collisions and other non-random properties for step-reduced sha-256. In *Selected Areas in Cryptography*, pages 276–293. Springer.

NIST (2007). The sha-3 cryptographic hash algorithm competition.

Rijmen, V. and Oswald, E. (2005). Update on sha-1. In *Topics in Cryptology–CT-RSA 2005*, pages 58–71. Springer.

Wang, X., Yin, Y. L., and Yu, H. (2005). Finding collisions in the full sha-1. In *Advances in Cryptology–CRYPTO 2005*, pages 17–36. Springer.

A True Random Number Generator based on quantum-optical noise

André de Almeida Ruegger[1], Geraldo A. Barbosa[2], Jeroen van de Graaf[3], Gilberto Medeiros[1], Julio Cezar de Melo[4], Roberto Nogueira[1], Wagner Rodrigues[1], Fernando Soares[1]

[1]Departamento de Física, UFMG

[2]QuantaSec—Consulting, Projects and Research in Physical Cryptography Ltd.

[3]Departamento de Ciência da Computação, UFMG

[4]Departamento de Engenharia Eletrônica, UFMG

Contact author: geraldoabarbosa@gmail.com

Abstract. *In this note we report a project to construct a True Random Number Generator based on quantum noise (shot noise of light). It achieves speeds of 1Gb/s and above which is about 3 orders of magnitude faster than the known bit generator Quantis (from IDQuantique). This generator is part of a platform for secure communications called KeyBITS.*

1. Motivation and problem

Suppose you have a bundle of fiber optics cable consisting of several hundred channels connecting two end points. You control the physical security of these end points, but you fear that the cable has been intercepted (maybe by some foreign spying agency). What can you do to protect your communications?

Here, quantum cryptography can be of help. The security of protocols such as BB84 [1], which can be run over fiber optics cables, is based on a principle of quantum mechanics, called the inference-disturbance principle[3]. It says that if an adversary is able to obtain information about bits sent over the channel, then the legitimate parties can quantify how much of the secret is leaked. This makes the quantum channel tamper proof, so it can be used to transport random bits to be used as a one-time pad key. Alternatively one can use this quantum channel for key transportation, and use symmetric encryption algorithms in order to encrypt all traffic. However, the optical technology used in quantum cryptography is very delicate since the protocol requires single photon pulses, which are both difficult to produce and to detect. For a thorough overview of this technology including all its problems, see [2].

A different approach, called mesoscopic optics, uses medium-intensity light of ≈ 100 to 10000 photons per pulse, which is still well below the high-intensity pulses used in conventional telecom systems ($> 10^{15}$photons/sec). Contrary to single-photon pulses, medium-intensity light can be produced and detected using off-the-shelve products of low cost, giving a significant advantage over BB84 and similar protocols. However, the underlying quantum-mechanical description of mesoscopic optics is different, and so is the underlying physical principle on which the security of the KeyBits Platform is based.

By sharing a preliminary seed between sender and receiver, the receiver know which basis to use in order to perform the right measurement and can therefore distinguish perfectly between a 0 and 1. But the adversary, who does not have this additional information, does not know how to measure correctly. The signal he measures will not be pure because of the shot noise or quantum-optical noise. In this context this noise manifests itself as an uncertainty in the phase angle as measured by the adversary, it is of an intrinsic quantum-mechanical nature, *not* an imperfection of the measurement equipment. This fact puts the adversary at a disadvantage compared to legitimate parties, leading to high imprecision when it tries to distinguish between a 0 or 1 sent over the channel.

Another manifestation of quantum-optical noise in a coherent field is the well-known fact that the amount of photons sent is not constant, but will always fluctuate, following a Poisson distribution. This fluctuation can be used as a source of true randomness, which is exactly what we use to construct the TRNG. For more details on shot noise see [**?**].

2. A True Random Number Generator

As a first step towards implementing the tamper-proof protocol outlined above, we focussed on implementing a True Random Number Generator (TRNG). (Actually it is a *bit* generator but we conform to traditional terminology.) In order to be useful for an optical channel one needs a true random bit generator at speeds compatible with fiber-optical technology, say 1 gigabit per second or more. This is a challenging problem since most TRNGs are very slow and have difficulties reaching even 1 megabit per second. Most designs are based on chips leading to PRNGs (Pseudo Random Number Generators) that may use thermal light fluctuations to enhance the randomness characteristics. But as part of the noise created in this process is of a thermal nature, it implies high correlations within the light field. For instance, the state-of-the-art TRNG implemented by Intel as a special instruction uses the physical randomness to create a random seed, which is then expanded using the AES cipher in order to obtain high speeds [4].

We are in the process of constructing a TRNG which uses quantum optics as the basis for randomness, by exploiting the quantum-optical noise of a laser in a coherent state.

3. Schematics of our quantum-optical TRNG

Fig. 1 shows a diagram of the TRNG. Basically, a laser excites a detector that produces a fluctuating current as its output. Upon amplification the voltage output presents fluctuations around an average voltage. After a signal processing stage, the $\pm$ fluctuations will be coded as V_+ and V_- signals that will correspond to the stream of bits K. Fig. 2 shows an extended view of a bench prototype of the TRNG. At the fundamental level the laser is in a coherent state in which the photon statistics has a Poissonian distribution in the number of photons n:

$$p(n) = \frac{e^{\langle n \rangle} \langle n \rangle^n}{n!} \, . \tag{1}$$

The most important characteristics of this distribution is that the photons present no correlation among themselves: $\langle n_1 n_2 \rangle = \langle n_1 \rangle \langle n_2 \rangle$. Similarly, a small group of photons

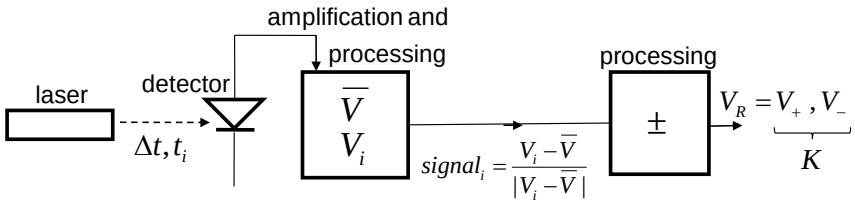


Figure 1. Schematics of the TRNG.

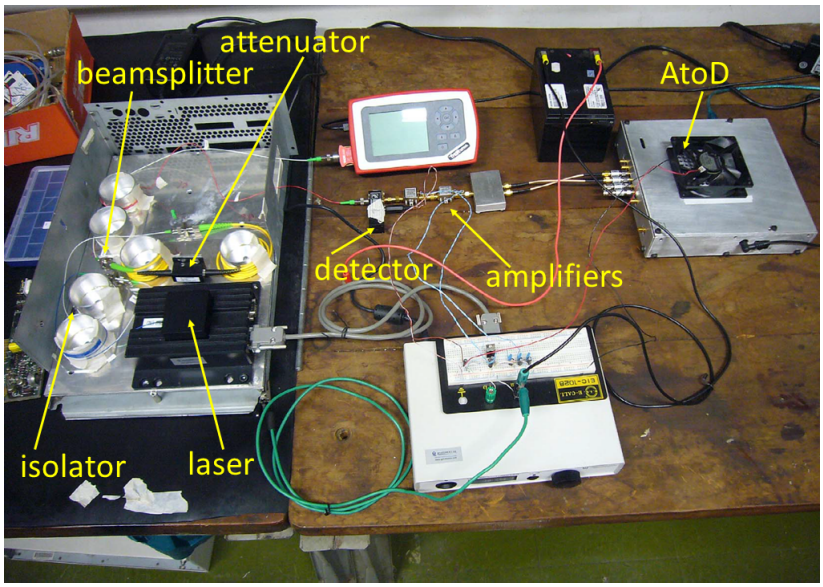
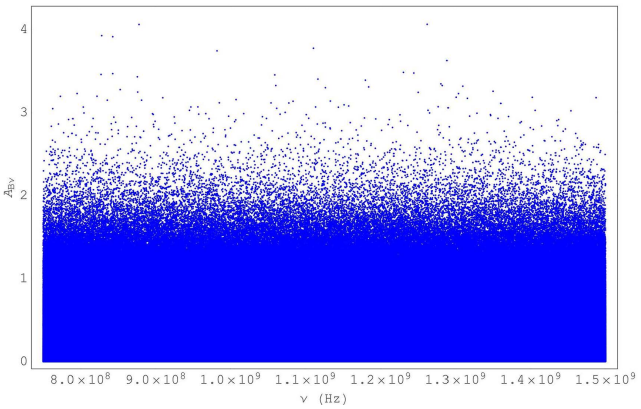


Figure 2. Bench prototype of TRNG.

measured within a short time interval Δt is independent of another group and this leads to the generation of uncorrelated bits. To achieve this characteristics the laser and detection system has to be prepared such that the noise associated to the optical field is much larger than all electronic noises present. This is known as "shot-noise" limited state.

Under this condition, bits are generated from the V_+ and V_- signals at the detector output. The average value of photons within Δt, $\langle n \rangle$, will lead to an average voltage $\overline{V}$. Δt is set such that $\Delta t \ll \tau$, the laser coherence time. This assures that the laser is with a given phase and, therefore, amplitude fluctuations are maximized. Photon number and phase have associated operators that do not commute, similarly as complementary pairs in a Heisenberg uncertainty principle.

The fluctuations around this value produce the bits. As an example, a sample of 19660800 bits were generated and its Fourier spectrum taken. Fig. 3(left-side) shows the "white-noise" characteristics of the Fourier spectrum of 19660800 bits. The right side shows results of the NIST tests on the 19660800 bits. All tests are plenty satisfactory. There is no parallel worldwide of a physical random generator with such functional simplicity and speed. The actual speed, $1.5GHz$ (bits/per second) is just due to the AtoD digitizer used and much higher speeds can be achieved with a faster electronics. Fundamentally, the optics field fluctuations has a white-noise characteristic for all light frequencies. Therefore, our scheme is not bounded by the physical principle used but just by the electronics – that can be improved according to the current state of art.



Teste	P-Value
ApproximateEntropy	0.812035
BlockFrequency	0.751001
CumulativeSums	0.784649
CumulativeSums(reverse)	0.842644
FFT	0.669945
Frequency	0.607710
LinearComplexity	0.709032
LongestRun	0.948537
NonOverlappingTemplate	146/148
OverlappingTemplate	0.672652
RandomExcursions	8/8
RandomExcursionVariant	18/18
Rank	0.695808
Runs	0.511987
Serial	2/2
Universal	0.953639

Figure 3. Left-side: "White-noise" spectrum of 19660800 bits. Right side: Results of the NIST tests on the 19660800 bits.

4. Conclusions

An essential part of a plataform for secure communication, a fast TRNG, has been constructed. It has an original and simple design and besides the use within this platform, it presents many possible independent uses, such as games. Its speed is high, compatible with current optical communication hubs and can still be increased with no fundamental bounds besides techological ones. The presented TRNG can follow those advances with no fundamental obstacle.

References

[1] C. Bennett, G. Brassard. *Quantum cryptography: Public key distribution and coin tossing.* In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984.

[2] V. Scarani, H. Bechmann-Pasquinucci, N. Cerf, M. Dusek, N. Lutkenhaus, M. Peev. *The Security of Practical Quantum Key Distribution.* `http://arxiv.org/abs/0802.4155`

[3] C. Fuchs. *Distinguishability and Accessible Information in Quantum Theory*. `http://arxiv.org/abs/quant-ph/9601020`.

[4] G. Taylor, G. Cox. *Digital randomness.* IEEE Spectrum, Vol. 48(9), pgs 32 – 58, September 2011

On Software Implementation of Arithmetic Operations on Prime Fields using AVX2

Armando Faz-Hernández, Julio López.

[1]Institute of Computing, University of Campinas.
{armfazh,jlopez}@ic.unicamp.br

Abstract. *AVX2 is the newest instruction set on Intel Haswell processor that provides simultaneous execution of operations over vectors of data. This work presents the advances on the applicability of AVX2 on the development of prime field arithmetic, which is a building block for the construction of Elliptic Curve Cryptosystems. Having as a goal the efficient and secure implementation of prime field arithmetic, we show some advantages that vector instructions offer compared against 64-bit implementations. In order to validate the results of our research, we present a benchmark obtained on a Haswell processor.*

1. Research context.

Our research is focused on the efficient implementation of prime field arithmetic, we aim to use the most efficient techniques that can benefit from the capabilities of the recent micro-architectures. Implementing prime field operations not only involves the correctness of operations, but also efficient and secure processing. The first goal is achieved by speeding up operations, extracting parallelism over data and/or using a special instruction set. Nonetheless, in order to meet the security requirement, the implementation requires a detailed information flow analysis, also to avoid secret-dependent code branching and avoiding calculations that could reveal fragments of secret data. In this work, we accomplish both requirements, the first one through the use of AVX2 vector instructions and the second one through the development of constant time execution code.

2. The vector instruction set: AVX2.

Observing the trend of contemporary processors, most of them have replicated execution units to accomplish with out-of-order execution, thus exploiting the instruction level parallelism present on programs. Another interesting trend on the micro-architectures design is the use of SIMD (Single Instruction Multiple Data) processing, i.e. processors are provided of vector instructions that simultaneously compute an operation on every element of vector registers. Haswell micro-architecture is an example of this trending, it includes sixteen 256-bit registers (hereafter referred as `YMM` registers) and is the first one to support the AVX2 vector instruction set.

The AVX2 set includes instructions mostly oriented to perform integer arithmetic operations, variable-shift on registers and permutations of 64-bit words between registers. The release of AVX2 extends most of the integer arithmetic from 2 to 4 simultaneous operations per instruction. Instructions for integer arithmetic are so attractive for the implementation of prime field arithmetic, where usually the size of operands implies the use of multi-precision arithmetic, i.e. the size of operands is greater than the size of the native word machine (nowadays 64 bits).

3. Prime field arithmetic.

Prime fields are denoted as $\mathbb{F}_p$ where p is a prime number. Usually, the elements of $\mathbb{F}_p$ are represented by the integers in the set $\{0, 1, \ldots, p-1\}$. Addition (ADD), subtraction (SUB) and multiplication (MUL) of elements are performed modulo p. Modular multiplication is processed in two steps: first, the integer multiplication (iMUL) of both inputs is computed, and secondly a modular reduction (MOD) is performed. The special case of integer multiplication when both inputs are equal, it is known as integer squaring (iSQR).

We focus on the application of prime fields for the construction of Elliptic Curve Cryptography (ECC) schemes. ECC is well known to provide stronger security with shorter key lengths when compared to the RSA cryptosystem. Recently, new proposals for selecting parameters of elliptic curves and prime fields were published, such as [Bernstein 2006, Bos et al. 2014, Aranha et al. 2013]. These proposals claim that such new parameters will accelerate the execution performance of prime field operations. Table 1 shows the prime fields recently proposed and also the prime field currently used in standardized ECC by NIST[1] [Gallagher et al. 2009].

4. The radix-R representation.

Here is presented an efficient representation of prime field elements, called *radix*-64. In order to understand what *radix*-R is, first we will show two examples of commonly used representations:

1. The size of primes is always greater than 64 bits, which is the size of registers in commodity processors. We can use an array of 64-bit words to store elements of the prime field. This kind of approach is commonly used in multi-precision mathematical libraries and is also known as *radix-64* representation.
2. In [Bernstein 2006], author proposes the use of *radix-25.5*, for which an element $A \in \mathbb{F}_p$ is represented by the following polynomial: $A(x) = \sum_{i=0}^{k-1} a_i x^i$ where k is the number of floating point registers used to represent that element and each a_i is bounded according to the precision of floating point registers.

These representations can be generalized to *radix*-R representation, thus an element $A \in \mathbb{F}_p$ is represented by the following polynomial: $A(x) = \sum_{i=0}^{k-1} a_i x^i$ where $a_i \in [0, 2^R)$ are integer coefficients and $k = \left\lceil \frac{\lg(p)}{R} \right\rceil$ is the number of R-bit words used to represent A. Now, we will describe the algorithms used to compute prime field operations using *radix*-R representation:

- **Addition/Subtraction.** Given two elements A and B on radix-R representation we can compute $C = A \pm B$ as $c_i = a_i \pm b_i$ for $i \in [0, k)$. Notice that these operations are totally independent and admit a parallel processing.
- **Integer multiplication.** It computes an intermediate result $C_{i+j} \leftarrow \sum a_i b_j$ for $i, j \in [0, k)$, here k^2 word multiplications are processed. These operations have no carry dependencies between them.
- **Modular reduction**. When pseudo-Mersenne primes are used ($p = 2^m - c$), modular reduction only requires to process $C_i = C_i + cC_{i+k}$, for $i \in [0, k)$. Notice that for these primes modular reduction can be done faster than for the NIST's primes.

[1]NIST stands for National Institute of Standards and Technology.

5. Efficient implementation using AVX2.

As one can see, we can benefit from the parallelism presented on the operations . Now, we will present an efficient and secure implementation of prime field arithmetic in *radix-*R representation using AVX2 instructions. A similar work of this implementation is found in [Bernstein and Schwabe 2012], where NEON vector instructions were used to accelerate cryptographic primitives using an ARM architecture.

Since a `YMM` register stores four 64-bit words, our implementation uses $t = \lceil \frac{k}{4} \rceil$ `YMM` registers to store the integer coefficients of *radix*-64 representation. In order to compute modular addition, the AVX2 instruction set contain the `VPADDQ` (`VPSUBQ` for subtraction) instruction that computes four simultaneous 64-bit additions. However the last carry bit of each addition is lost. In order to overcome this issue, we restrict the R parameter to be $R < 64$, so each 64-bit operation has at least an extra available bit to store the carry bit produced by the addition operation. This restriction also applied to the case of integer multiplication. In order to compute $A \times B$, one has to add k intermediate products $a_i b_j$ for $i, j \in [0, k)$. To determine a bound for R we have:

$$\begin{aligned} k(2^R - 1)^2 &< 2^{64} \\ \log_2(k) + 2R &< 64 \\ \log_2(\log_2(p)) - \log_2(R) + 2R &< 64 \\ R - \tfrac{1}{2}\log_2(R) &< \tfrac{1}{2}(64 - \log_2(\log_2(p))). \end{aligned} \quad (1)$$

Then, the larger integer that holds (1) is $R = 30$, which nicely fits with the interface of `VPMULDQ` instruction. This instruction performs four simultaneous 32×32 bit multiplications. Finally, in the computation of the modular reduction, the terms cC_{i+k} are computed using shifts on vector registers instead of multiplications, and this can be done easily through the use of `VPSLLQ` and `VPSRLQ` instructions.

6. Preliminary results.

In the Table 1, we show the timings for the main operations on prime fields. The *radix-64* row refers to the implementation that uses native 64-bit instructions, such as a 64×64 bit multiplier (`MULX` instruction) and a 64-bit adder that computes addition with carry (`ADC` instruction). The results of our AVX2 implementation are shown in the *vec-radix-30* row.

We highlight that most of our timings using AVX2 instructions are competitive with the radix-64 implementation, for example, a modular multiplication (MUL) using the Curve25519's prime can be computed in 52 clock cycles on *radix-64*; while using the AVX2 implementation, it takes only 53 clock cycles, achieving almost the same performance. For the case of modular squaring (SQR), when is compared to *radix-64* implementation, our implementation is faster by 4 and 9 clock cycles for the Curve25519 and Curve1174 prime fields, respectively.

Modular addition and subtraction operations present almost the same performance 8-9 clock cycles. For vector implementation, we have that $R < 32$, this allows to compute more than one modular addition before a coefficient reduction be needed. The *coefficient reduction* is an operation that reduces each coefficient to the range $[0, 2^R)$ propagating the carries to the next significant coefficient.

			NISTp256		Curve25519		Curve1174	
			$\mathbb{F}_{2^{256}-2^{224}+2^{192}+2^{96}-1}$		$\mathbb{F}_{2^{255}-19}$		$\mathbb{F}_{2^{251}-9}$	
	iMUL	iSQR	ADD	MOD	ADD	MOD	ADD	MOD
radix-64	37	30	14	53	8	15	8	15
vec-radix-30	35	23	9	60	9	18	9	13

Table 1. Clock cycles measured to process each prime field operation on a Haswell processor Intel Core i7-4770.

The idea behind *radix-R* representation is to enable parallel computation that AVX2 vector instructions can take advantage. Our results show that the use of AVX2 is worthwhile on the implementation of prime field arithmetic. However, this representation also presents some side issues that results on additional operations, such as the coefficient reduction which takes around 28 clock cycles to be computed.

We keep investigating on optimization techniques for coefficient reduction and the application of the lazy reduction technique in order to minimize the impact of this modular operation. In order to compare our results against other implementations we will make a proof of concept on an elliptic curve cryptography protocol.

Acknowledgments: We would like to thank the anonymous reviewers for their helpful suggestions and comments. This research was supported in part by the Intel Labs University Research Office.

References

Aranha, D. F., Barreto, P. S. L. M., Pereira, G. C. C. F., and Ricardini, J. E. (2013). A note on high-security general-purpose elliptic curves. Cryptology ePrint Archive, Report 2013/647. `http://eprint.iacr.org/`.

Bernstein, D. and Schwabe, P. (2012). NEON Crypto. In Prouff, E. and Schaumont, P., editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 320–339. Springer Berlin Heidelberg.

Bernstein, D. J. (2006). Curve25519: New Diffie-Hellman Speed Records. In Yung, M., Dodis, Y., Kiayias, A., and Malkin, T., editors, *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228. Springer.

Bos, J. W., Costello, C., Longa, P., and Naehrig, M. (2014). Selecting Elliptic Curves for Cryptography: An Efficiency and Security Analysis. Cryptology ePrint Archive, Report 2014/130. `http://eprint.iacr.org/`.

Gallagher, P., Foreword, D. D., and Director, C. F. (2009). FIPS PUB 186-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION Digital Signature Standard (DSS).

Autenticação contínua para smartphones baseada em assinatura acústica

Marcelo da Luz Colomé, Raul Ceretta Nunes

Departamento de Computação Aplicada – Centro de Tecnologia

Universidade Federal de Santa Maria, Santa Maria – RS – Brazil

{marcelocolome, ceretta}@inf.ufsm.br

Abstract. *With the increasing of data and sensible information stored in smartphones, control the access to this devices is essential in order to mitigate risks. In this sense, a variety of authentication mechanisms has been explored, as the use of passwords and gestures. However, users tend to set weak password combinations or gestures that are easy to reproduce. This fact has been stimulating the research for continuous authentication methods, based on the user's interaction, which also run in background. The purpose of this paper is a new continuous authenticating method based on acoustic signature that is produced by the user-smartphone's interaction.*

Resumo. *Com o aumento de dados e informações sensíveis armazenados nos smartphones, controlar o acesso aos dispositivos é essencial para redução de riscos. Neste sentido diferentes mecanismos de autenticação tem sido explorados, tal como o uso de senhas ou gestos. Entretanto, os usuários costumam utilizar combinações ou gestos facilmente reproduzíveis, o que têm estimulado a perquisa por métodos de autenticação contínua baseados na interação do usuário e que executam em background. Este artigo propõe um novo método de autenticação contínua baseado em assinatura acústica produzida a partir da interação usuário-smartphone.*

1. Introdução

No cenário atual, *smartphones* são largamente utilizados, sendo estes dispositivos capazes de armazenar dados e informações importantes (sensíveis). Porém, estes dispositivos estão suscetíveis a uma variedade de riscos, tais como perda, roubo ou invasões, o que pode resultar em um acesso não permitido aos dados e informações. Dentre os principais tipos de autenticação de smartphones estão a autenticação por senha e a autenticação baseada em gestos, mas segundo [Frank et al. 2013] os usuários geralmente definem senha fracas, ou até mesmo desativam este tipo de proteção, deixando o dispositivo suscetível a ataques. Isto é um claro indicativo que apenas este tipo de autenticação não oferece uma boa solução para resolver a autenticação em smartphones.

Uma técnica de autenticação alternativa, chamada de autenticação contínua (*continuous authentication*), consiste em verificar continuamente, em background, se o usuário é autentico, oferecendo uma segunda barreira de proteção em complemento à

autenticação convencional [Frank et al. 2013]. Sem interromper o usuário, este tipo de autenticação costuma ser baseada em dados biométricos que podem ser captados do usuário que interage com o dispositivo pelos sensores presentes no mesmo. No caso dos *smartphones*, estas informações podem ser captadas através da tela sensível ao toque, ou com sensores presentes em grande parte dos *smartphones* atuais como o giroscópio, o acelerômetro, o magnetômetro e o microfone. Uma vez captadas, as informações podem ser usadas para a criação de um perfil individual de cada usuário, uma espécie de assinatura biométrica que é usada para identificar, de uma maneira não obstrutiva, se o usuário do sistema é autêntico.

Alkilani e Shirkhodaie (2013) propuseram uma técnica de reconhecimento de assinaturas acústicas para interações humano-objeto e que foi aplicada em sistema de vigilância. O desafio foi identificar o padrão de som (assinatura) emitido pela manipulação de objetos de diferentes materiais, como metal, plástico e madeira, em diferentes situações, mas a solução é computacionalmente complexa. No caso dos smartphones, oferecer um novo método de autenticação contínua baseado no som proveniente da interação humana com o dispositivo ainda é uma tarefa não resolvida. Há de se considerar que os smartphones são dispositivos com limitações de processamento e de consumo de energia, o que demanda soluções específicas.

Este trabalho apresenta uma solução para a autenticação contínua em smartphones, a qual é baseada no processamento em background dos sons capturados pelo microfone do dispositivo e na identificação de uma assinatura acústica que permita detectar o uso por usuário legítimo ou por usuário não autorizado (intruso).

2. Solução Proposta

Visando solucionar o problema da autenticação contínua baseada em sons provenientes da interação usuário-dispositivo, este trabalho alia estratégias já aplicadas à autenticação contínua com técnicas usadas para a identificação de padrões de som (assinatura acústica).

Na autenticação contínua, diferentes técnicas de seleção de atributos são utilizadas para reduzir o espectro das informações de interesse. Alguns autores como Song et al. (2013) preferem a técnica Fisher de seleção de atributos [Duda e Stork 2001], uma das técnicas supervisionadas mais utilizadas devido ao seu bom desempenho em geral, enquanto outros como Govindarajan, Gasti e Balagani (2013) utilizam a seleção não supervisionada baseada na derivação da mediana absoluta (MAD). Por não utilizar informação de referência, a seleção não supervisionada pode facilitar a escalabilidade por não exigir que o processo de seleção de atributos seja refeito quando um novo usuário do dispositivo é considerado, tal como pode acontecer em dispositivos empresariais.

Para a fase de seleção de atributos, considerando que os dados provenientes do microfone podem resultar numa classe de dados não balanceada, neste trabalho optou-se pela utilização dos algoritmos *Random Forests* [Liaw and Wiener 2002] e *k-d-tree* para classificar os atributos mais significativos. Baseado nesta classificação, um perfil do usuário pode ser construído a partir das informações extraídas do áudio proveniente da interação do usuário com o *smartphone*.

A obtenção de uma assinatura acústica depende de técnicas de identificação de padrões de som similares as usadas para a extração de atributos dos arquivos de áudio, tal como a Transformada de Fourier (*Fast Fourier Transform – FFT)* usada em [Alkilani and Shirkhodaie 2013]. Por isto, o primeiro passo do nosso método é a extração dos atributos de áudio presentes nos arquivos que podem ser gravados a partir do microfone padrão do *smartphone*.

A primeira interação do usuário é usada para o treinamento do modelo de autenticação e subsequente para teste. O processo de extração é seguido do processo de seleção, que tem como objetivo identificar os atributos mais discriminativos, isto é, os que irão proporcionar uma maior chance de obtenção de padrões de som significativamente distintos.

Na fase de extração, o arquivo de áudio gerado durante a interação do usuário com o *smartphone* é dividido em pequenos pedaços contendo apenas os momentos sonoros onde há informação relevante, tal como em [Alkilani and Shirkhodaie 2013]. A técnica empregada nesta etapa é determinar quando ocorrem os toques na tela do smartphone através do sensor de toque existente no dispositivo [Frank et al. 2013], pois estes toques geram eventos, como quando uma ligação está sendo efetuada pelo usuário. Posteriormente então, divide-se o arquivo em várias partes baseado na informação temporal dos eventos captados. Assim a tarefa de determinar o momento exato deste tipo de interação é facilitada, necessitando apenas o ajuste de sincronia do arquivo de áudio com os dados provenientes do sensor de tela. Um script que lê as informações de quando os toques ocorreram no dispositivo, divide o arquivo principal em arquivos menores que contém apenas o som proveniente da interação do usuário com este dispositivo. Após esta etapa, o *software* JAudio [JAudio 2013] é usado para a extração de valores de domínio de frequência através da Transformada Fourier. A partir deste valor encontrado para cada pequeno arquivo de áudio calcula-se o valor mínimo, máximo, médio, desvio padrão e mediana. Todos estes valores farão parte do vetor que representa o perfil do usuário.

Construído o perfil do usuário, a próxima etapa é a comparação dos perfis de usuários. Usa-se a distância Manhattan e a distância Euclideana para comparar a distância entre os vetores que representam os perfis de usuário (construídos na etapa anterior). Compara-se assim os perfis do próprio usuário consigo mesmo e os perfis de usuário entre si. Quanto menor a distância entre o próprio usuário e maior a distância entre os usuários diferentes, menor é a probabilidade de autenticação com falsa aceitação (*False Acceptance Rate*) e falsa rejeição (*False Rejection Rate*) do modelo de autenticação [Govindarajan, Gasti and Balagani 2013].

O produto final do método de autenticação proposto é um modelo de autenticação que pode ser aplicado a um *smartphone* convencional, sem a necessidade de implantação de *hardware* adicional, e que permite verificar continuamente o usuário baseado no som produzido pela sua interação com o dispositivo, isto é, sua assinatura acústica. Quando o sistema de autenticação está ativo no *smartphone*, ele capta o áudio proveniente da interação e o compara com a assinatura acústica previamente gravada no dispositivo. Portanto, se a assinatura acústica não corresponder com o perfil acústico do usuário pode-se enquadrar o usuário como intruso e a autorização de acesso pode ser negada.

É importante ressaltar que os testes estão em desenvolvimento e que a precisão do método ainda não foi computada. Além disto, salienta-se que o método deve ser utilizado em conjunto com outros métodos de autenticação para melhor garantia na cessão de diretos de acesso, dado que os métodos de autenticação contínua não substituem métodos de autenticação convencionais.

3. Considerações finais

Conforme exposto neste trabalho, autenticação de smartphones requer uma abordagem diferente das técnicas utilizadas na segurança de outros dispositivos como os computadores pessoais. A autenticação contínua oferece esta abordagem diferenciada, na qual o usuário é autenticado com base em seu comportamento, e não na informação que ele carrega consigo, como uma senha ou um padrão de gestos. A autenticação contínua baseada em sons produzidos pela interação entre o usuário e o dispositivo é uma técnica promissora de autenticação contínua, porém ao que alcança o conhecimento dos autores, é algo ainda não explorado fora deste trabalho.

A solução apresentada neste trabalho utiliza técnicas de extração de atributos de som, técnica usada com sucesso em [Alkilani and Shirkhodaie 2013] para a elaboração de um perfil de sons provenientes da interação humano-objeto, juntamente com estratégias de autenticação contínua usadas em [Frank et al. 2013], [Serwadda and Phoha 2013], [Govindarajan, Gasti and Balagani 2013], [Song et al. 2013].

4. Referências

Frank, M., Biedert, R., Ma, E., Martinovic, I. and Song, D. (2013). Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions On Information Forensics And Security*, 8(1):136-148, January.

Alkilani, A. and Shirkhodaie, A. (2013). Acoustic signature recognition technique for Human-Object Interactions (HOI) in persistent surveillance systems. *Proc. SPIE 8745, Signal Processing, Sensor Fusion, and Target Recognition XXII*, May, doi:10.1117/12.2018627.

Serwadda, A. and Phoha, V. V. (2013). When kids toys breach mobile phone security. *ACM SIGSAC conference on Computer & communications security (CCS '13)*.

Govindarajan, S., Gasti, P. and Balagani, K. S. (2013). Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data. *Biometrics: Theory, Ap- plications and Systems (BTAS), IEEE Sixth International Conference on*, vol., no., pp.1-8, doi: 10.1109/BTAS.2013.6712742.

JAudio 1.0 (2013). http://jaudio.sourceforge.net, Acesso em Julho.

Song, Y., Salem, M. B., Hershkop, S. and Stolfo, S. J. (2013). System level user behavior biometrics using fisher features and gaussian mixture models. *Security and Privacy Workshops (SPW), 2013 IEEE*, pages 52-59, May.

Duda, P. E. H. R. O. and Stork, D. G. (2001). Pattern Classication. Wiley-Interscience Publication.

Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. R News, v.2, n.3, December, pp.18-22.

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

WTICG – VIII Workshop de Trabalhos de Iniciação Científica e de Graduação

Análise de Segurança de Conversores Serial-Ethernet e Microcontroladores Tibbo

Ildomar Gomes de Carvalho Junior[1], **Rafael R. Obelheiro**[1]

[1]Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC)
Caixa Postal 631 – 89223-100 – Joinville – SC – Brazil

ildomarcarvalho@gmail.com, rafael.obelheiro@udesc.br,

Abstract. *Microcontrollers and serial-to-Ethernet converters are used in industrial control systems for communication and control of devices. The malfunctioning of these components can result in damage both in equipments and in the manufacturing process of products, bringing risks to the physical integrity of people who operate machines that depends of these embedded systems. The growing integration of microcontrollers and serial-to-Ethernet converters to the Internet and the increase of vulnerabilities involving industrial control systems make the security of these devices become a more and more important feature. This article presents the use of fault injection techniques through the network to evaluate the security of the TCP/IP stack of the microcontroller and serial-to-Ethernet converter Tibbo EM1206. The procedure followed for this evaluation can be used to conduct similar evaluations.*

Resumo. *Microcontroladores e conversores serial-Ethernet são utilizados em sistemas de controle industrial para a comunicação e controle de dispositivos. O funcionamento incorreto destes componentes pode acarretar danos tanto nos equipamentos como no processo de manufatura de produtos, trazendo riscos até mesmo para a integridade física de pessoas que operam máquinas que dependem desses sistemas embarcados. A crescente integração de microcontroladores e conversores à Internet e o aumento de vulnerabilidades envolvendo sistemas de controle industriais fazem com que a segurança desses dispositivos se torne um atributo cada vez mais importante. Este artigo apresenta o uso de técnicas de injeção de faltas através da rede para avaliar a segurança da pilha TCP/IP do microcontrolador e conversor serial-Ethernet Tibbo EM1206. O processo seguido nesta avaliação pode ser utilizado para a condução de avaliações similares.*

1. Introdução

Sistema de Controle Industrial, ou SCI, é um termo que engloba diversos tipos de sistemas computacionais, como produção e distribuição de energia elétrica, fornecimento e tratamento de água, produção e distribuição de petróleo e combustíveis e na telecomunicação [Ralston et al. 2007]. Microcontroladores e conversores serial-*Ethernet*, são exemplos de dispositivos encontrados em SCIs.

Um microcontrolador é um dispositivo que contém processador, memória e dispositivos de entrada e saída em um único *chip*. É utilizado por exemplo para controle de

motores, monitoramento de equipamentos e robótica industrial [Brudna 2000]. Já conversores serial-*Ethernet* são utilizados em equipamentos que só possuem comunicação serial e precisam comunicar-se em rede, como por exemplo em servomotores.

SCIs podem estar conectados à *Internet* e por isso é necessário assegurar o nível de segurança dos seus dispositivos. Ataques via rede a estes sistemas podem causar defeito na fabricação de produtos ou mesmo danos e até a perda completa de máquinas que são operadas por meio destes. Por exemplo, em um servomotor conectado a um destes dispositivos, o atacante poderia alterar os parâmetros de configuração, enviar comandos não autorizados ou causar seu travamento. Visto que este tipo de motor é inserido em máquinas que necessitam de alta precisão como máquinas CNC (Controle Numérico Computarizado), erros e falhas podem levar à perda do produto fabricado, da máquina na qual ele está inserido ou até oferecer riscos à integridade física de quem a opera.

Para fornecer um estudo de caso sobre os problemas encontrados em dispositivos comumente utilizados na indústria, este trabalho analisou a robustez do microcontrolador e conversor serial-*Ethernet* EM1206 fabricado pela Tibbo[1], buscando vulnerabilidades que violem as propriedades de segurança, a saber, integridade, confidencialidade e disponibilidade [ISO/IEC 2008], por meio de injeção de ataques. A análise revelou que o dispositivo estudado é suscetível a travamentos provocados por tráfego de rede, além de possuir algumas vulnerabilidades que podem permitir que usuários não autorizados o acessem de forma indevida. Além das vulnerabilidades em si, o artigo também descreve a metodologia usada na avaliação, que pode servir de base para estudos similares.

O texto está organizado em 5 seções. Na seção 2 será dada uma introdução sobre sistemas de controle industrial, ambiente onde os módulos Tibbo são utilizados. Na seção 3 é apresentada a técnica de injeção de faltas e ataques, utilizada neste trabalho. Na seção 4 será apresentado o ambiente de testes, a metodologia adotada e os resultados. Por fim, nas seções 5 e 6 são apresentadas uma discussão dos resultados obtidos e conclusões.

2. Sistemas de Controle Industrial

SCIs são sistemas computacionais usados em vários setores da indústria para automatizar processos industriais [Melton et al. 2004], onde é necessário o constante monitoramento de dados através de sensores e correção ou alteração dos processos em execução através de seus atuadores.

Dependendo da aplicação em que um SCI é utilizado, pode ser necessário centralizar os dados de dispositivos que encontram-se distantes geograficamente. Visto que estas informações terão de ser transmitidas através da Internet ou de uma rede privada, a preocupação com a segurança destes dados é grande [Hart 2004]. A Figura 1 mostra um exemplo de SCI, onde é necessária a transmissão de dados pela Internet.

[1]http://www.tibbo.com/

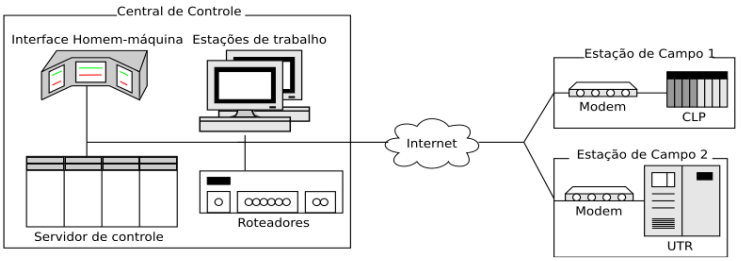


Figura 1. Exemplo de um SCI, adaptado de [Stouffer et al. 2007].

Por tratar-se de um microcontrolador e conversor serial-*Ethernet*, o EM1206 pode ser utilizado tanto para controle de atuadores, quanto para monitoramento de sensores. Por possuir comunicação *Ethernet*, ele também pode ser utilizado em SCIs para transmissão de dados e comandos tanto em uma rede local quanto através da Internet, o que pode representar um problema caso um atacante explore suas vulnerabilidades.

3. Injeção de Faltas e Ataques

Considerando que os módulos Tibbo são dispositivos disponíveis comercialmente e que não se tem acesso ao seu código fonte, foi necessário conduzir um estudo experimental, através da técnica de injeção de faltas. Esta é uma técnica experimental que consiste em provocar faltas deliberadas em um sistema sob teste e observar se as faltas injetadas fazem com que o funcionamento do sistema desvie de sua especificação [Arlat et al. 1990], [Clark and Pradhan 1995], [Hsueh et al. 1997].

O propósito da injeção de faltas é avaliar as propriedades de confiança no funcionamento (*dependability*) [Avizienis et al. 2004] do sistema. Embora a confiança no funcionamento tenha uma intersecção significativa com a segurança [Avizienis et al. 2004], o uso de injeção de faltas estava inicialmente focado em propriedades de confiabilidade e disponibilidade, e não em integridade ou confidencialidade. Quando a injeção de faltas é aplicada em segurança, ela pode ser chamada de injeção de ataques [Antunes et al. 2005].

A injeção de faltas permite testar sistemas após a fase de desenvolvimento, sejam sistemas em sua versão final ou protótipos [Arlat et al. 1990]. Seu uso permite complementar outras técnicas, como modelagem analítica de confiabilidade, para obter uma validação mais completa [Clark and Pradhan 1995]. A análise dos resultados observados em um estudo de injeção de faltas permite compreender o comportamento do sistema em situações adversas e avaliar a eficácia de seus mecanismos de proteção e recuperação, além de ajudar a identificar eventuais deficiências que permitam que o sistema falhe (o que é primordial para que se possa corrigi-las) [Clark and Pradhan 1995], [Hsueh et al. 1997].

Neste trabalho foram injetadas faltas nos dispositivos Tibbo através da rede, enviando tráfego malicioso para avaliar a robustez e a segurança de sua pilha TCP/IP. A técnica de injeção de faltas escolhida para este trabalho foi *Fuzzing*, a qual tem por objetivo expor falhas em aplicações inserindo nelas entradas mistas de dados válidos e inválidos. Esta técnica requer três operações básicas: gerar entradas aleatórias, injetar essas entradas na aplicação e, por fim, observar se esta entrada causou algum tipo de falha à aplicação [Banks et al. 2006]. Este fluxo é apresentado pela Figura 2.

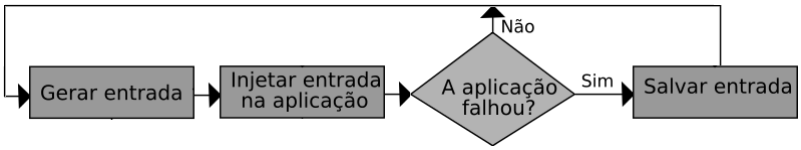


Figura 2. Fluxograma básico das iterações de um *fuzzer*, adaptado de [Oehlert 2005].

As entradas exploradas neste trabalho foram a estrutura e os parâmetros dos protocolos de rede. Estes são especificados através das *Request for Comments*, ou RFCs, que são documentos considerados padrão. Mesmo com estas especificações disponíveis na *Internet*, desenvolvedores podem cometer erros durante a codificação da pilha TCP/IP, e é nestas possíveis falhas que os *fuzzers* deste trabalho são focados, explorando um grande conjunto de possíveis entradas inválidas ou semi-válidas, para descobrir quais delas produzem comportamentos indesejados [Gu et al. 2011].

4. Avaliação Experimental

4.1. Ambiente de Testes

Para o desenvolvimento dos testes deste trabalho, as seguintes ferramentas de busca de vulnerabilidades e de análise de pacotes de rede foram utilizadas:

1. Nmap [Lyon snt], para varredura de portas e detecção do sistema operacional;
2. Scanner de vulnerabilidades Nessus [Anderson 2003];
3. Wireshark [Combs snt], para análise dos pacotes de rede;
4. IP *Stack Integrity Checker* [Xiao and Frantzen snt], o qual é um conjunto de ferramentas para executar *fuzzing* na pilha TCP/IP. Suas ferramentas utilizadas foram o ESIC, que gera *frames Ethernet*; ISIC que gera pacotes IP; TCPSIC que gera pacotes TCP; e UDPSIC que gera datagramas UDP.
5. BED, ou *Bruteforce Exploit Detector* [Damaye 2012], para realizar testes *fuzzing* na aplicação *web* de gerenciamento do módulo.
6. Scapy [Biondi 2007] para criar *scripts* para reproduzir as vulnerabilidades encontradas.

O ambiente de testes utilizado é o apresentado pela Figura 3 e o EM1206 estava trabalhando com a *firmware* de conversor serial-Ethernet.

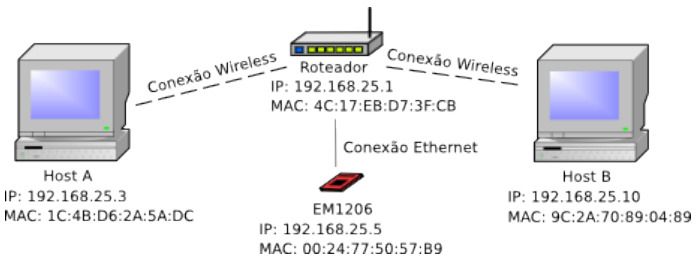


Figura 3. Cenário utilizado nos testes deste trabalho.

4.2. Metodologia Adotada

A metodologia adotada para este trabalho foi inspirada na metodologia proposta por Pothamsetty e Balinsky (2003), analistas da Cisco Systems, adotando as etapas:

1. Reconhecimento, usando técnicas para identificar remotamente o sistema operacional e descobrir quais serviços de rede estão ativos no sistema alvo;
2. Análise por camadas, onde foram executadas injeção de pacotes malformados, ataques de negação de serviço e pesquisa de vulnerabilidades com o *software* Nessus nos protocolos *Ethernet*, IP, UDP, TCP e HTTP;

4.3. Resultados

Nas próximas seções são apresentados os resultados da etapa de reconhecimento e as vulnerabilidades encontradas nas camadas *Ethernet*, IP, UDP, TCP e de aplicação.

4.3.1. Reconhecimento

As técnicas de reconhecimento têm por objetivo coletar informações sobre o dispositivo atacado [Pothamsetty and Balinsky 2003]. Neste trabalho, as técnicas de reconhecimento foram as de identificação do sistema operacional e a de varredura da pilha TCP/IP.

Ambas as técnicas de reconhecimento citadas foram executadas pelo Nmap e o resultado mostrou que o Nmap não conseguiu reconhecer o sistema operacional. A varredura de portas mostra três TCP portas abertas: porta 23 a qual, ao contrário do que o Nmap diz, não roda o serviço Telnet, mas uma aplicação de gerenciamento do módulo [Tibbo 2013]; porta 80, a qual roda o servidor *web*, também usado para gerenciamento; e a porta 1001, que é a porta utilizada para conversão serial-*Ethernet*. Esta última pode ser alterada pelo usuário para que seja a porta TCP ou UDP que ele desejar.

4.3.2. *Ethernet*

Foram realizados dois testes para realizar a busca de vulnerabilidades neste protocolo. O primeiro foi realizado com o *software* ESIC, onde este aplicou *fuzzing* em todos os campos possíveis do cabeçalho *Ethernet*, porém o EM1206 não mostrou-se vulnerável aos ataques *fuzzing*.

O segundo teste foi executado pelo Nessus, o qual indicou que o EM1206 possui a vulnerabilidade Etherleak. A RFC 1042 [Postel 1988] especifica um tamanho mínimo para os quadros *Ethernet*. Caso o conteúdo do quadro não preencha este tamanho mínimo necessário, devem ser adicionados zeros a ele até que o quadro seja preenchido completamente. A vulnerabilidade chamada Etherleak acontece quando os dados que preenchem esse espaço restante vêm de *buffers* do sistema [Arkin and Anderson 2003].

Foi verificado que, quando uma requisição ARP é feita ao Tibbo, dados do *buffer* de rede são acrescentados ao quadro *Ethernet*. A partir da captura destas requisições ARP, foi possível obter a senha do conversor, que foi configurada como **tst**. A Figura 4 mostra em destaque o momento desta captura.

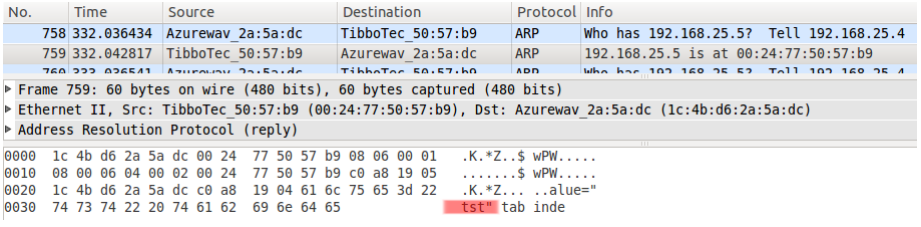


Figura 4. Captura da senha do EM1206, devido ao Etherleak.

4.3.3. IP e UDP

Para testar a robustez da implementação dos protocolos IP e UDP do módulo EM1206, foram utilizados separadamente os *softwares* ISIC e UDPSIC. O EM1206 não mostrou-se vulnerável aos ataques *fuzzing*, porém o fluxo de dados injetado por cada ferramenta acabou causando um ataque *flooding* ao dispositivo. Notou-se também que, após cessados os ataques, o dispositivo não voltou ao seu funcionamento normal, sendo necessário reiniciá-lo manualmente.

4.3.4. TCP

Para detectar vulnerabilidades no protocolo TCP, três testes foram executados. O primeiro foi feito com o *software* TCPSIC e o resultado foi similar ao encontrado na seção anterior, onde o módulo é vulnerável a *flooding*, porém não é vulnerável a *fuzzing*.

O segundo teste foi feito para verificar o quão eficiente é o gerador de números iniciais de sequência. Se um dispositivo em rede possui um gerador do número inicial de sequência com uma aleatoriedade fraca, é possível injetar dados em uma conexão TCP já estabelecida [Pothamsetty and Balinsky 2003]. Para avaliar o grau de aleatoriedade do gerador de ISN do EM1206, dois *softwares* foram utilizados: o Nmap e Nessus.

O Nmap avaliou a aleatoriedade do EM1206 como forte, porém o Nessus avaliou a aleatoriedade do gerador de ISN como fraca. Devido a esta divergência entre os resultados, foi necessário fazer uma análise manual dos pacotes para descobrir qual está correto. Para isso foi criado um *script* com o Scapy que envia segmentos SYN para o EM1206 e imprime o número de sequência utilizado na resposta SYN-ACK. Através deste *script* foi constatado que o gerador de números iniciais de sequência do EM1206 simplesmente incrementa seu valor em 64000 a cada 500 ms. Esse comportamento viola a RFC 6528 [Gont 2012], que especifica que os ISNs devem ser aleatorizados.

Apesar de o resultado gerado pelo Nessus ser o correto, um ponto a ser destacado sobre a análise feita por este *plugin* do Nessus é que ele pode gerar falsos positivos. Conforme pode ser visto em seu código fonte [Deraison 2002], o Nessus envia dois segmentos SYN, captura o número de sequência das respostas SYN-ACK e os compara: se eles forem iguais a aleatoriedade é dada como fraca, caso contrário ela é dada como forte.

O terceiro teste também foi feito pelo Nessus e encontrou uma vulnerabilidade onde é possível encerrar uma conexão já estabelecida entre o EM1206 e outro dispositivo enviando pacotes RST ilegítimos (*reset spoofing*). Para confirmar esta vulnerabilidade, foi criado um *script* utilizando a ferramenta Scapy, o qual envia um pacote com as *flags* SYN e RST para uma conexão TCP já estabelecida entre o Tibbo e um *host*, utilizando um número de sequência e um ACK com os valores errados.

Para este teste havia uma conexão TCP na porta 1001 do EM1206 entre este e o *host* A. A Figura 5 mostra a execução do *script* criado: o pacote 135 e os anteriores mostram o tráfego normal entre o *host* A e o EM1206; o pacote 227 mostra o momento

em que o *host* B envia o segmento RST para o EM1206 (o número de sequência utilizado é 0), com seu IP alterado para que parecesse ser do *host* A; o pacote 274 mostra o momento em que o *host* A envia mais um pacote legítimo ao EM1206; e por fim, o pacote 275 mostra o momento em que o EM1206 finaliza a conexão com o *host* A.

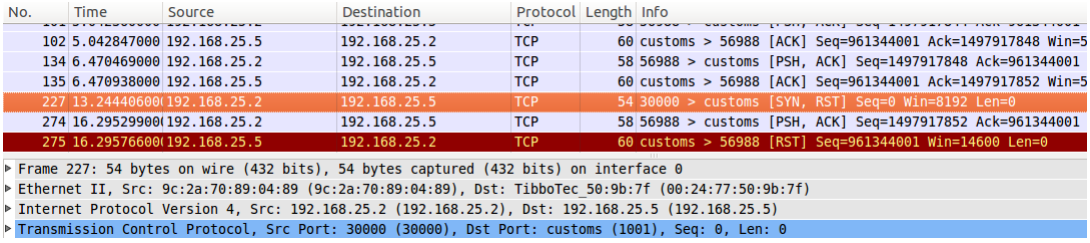


Figura 5. Captura do *reset spoofing* executado pelo *script*.

4.3.5. Aplicação

Na camada de aplicação foram encontradas quatro vulnerabilidades. A primeira é quanto à opção autocompletar do campo senha, na aplicação *web* do EM1206. Não desabilitar o autocompletar oferece um risco de perda de confidencialidade.

A segunda é uma vulnerabilidade descoberta através da ferramenta BED, a qual foi utilizada neste teste para avaliar o comportamento do servidor web embarcado do conversor EM1206 diante de ataques *fuzzing* na camada de aplicação. O EM1206 não mostrou-se vulnerável aos ataques *fuzzing*, porém o fluxo de dados gerado pelo BED causou um ataque *flooding*. Um fato interessante é que este ataque *flooding* funcionou apenas com os parâmetros *GET* e *POST*, não tendo sido efetivo com as linhas de cabeçalho HTTP.

A terceira é uma vulnerabilidade ao roubo do *cookie* de uma sessão estabelecida. Quando um usuário se autentica na página *web* do EM1206, o conversor dá a este usuário um *cookie* para identificar que ele está autenticado [Kurose and Ross 2006]. Este recurso traz consigo uma vulnerabilidade, pois caso um atacante consiga obter este *cookie*, ele pode usá-lo para obter a senha de administrador do EM1206.

Por fim, foi encontrada uma vulnerabilidade quanto ao envio não criptografado da senha do conversor. É possível acessar e alterar as configurações do conversor EM1206 da Tibbo de três maneiras: através da sua página *web* embarcada, do *software* DS Manager ou de uma aplicação que roda na porta TCP 23, a qual não é o Telnet padrão [Tibbo 2013]. Estes recursos possuem em comum uma mesma vulnerabilidade: eles enviam a senha pela rede de forma aberta, sem nenhum tipo de criptografia. O DS Manager possui uma falha ainda mais grave, pois a senha é enviada por *broadcast* pelas camadas de enlace e de rede.

5. Discussão dos resultados

Diversas vulnerabilidades foram encontradas durante o desenvolvimento de trabalho, sendo que elas variam bastante em níveis de gravidade ao sistema. A tabela 1 resume as vulnerabilidades apresentadas nas seções 4.3.2 a 4.3.5, separadas por camadas.

Dentre todas as vulnerabilidades encontradas, algumas merecem um destaque especial. Etherleak foi capaz de mostrar a senha do EM1206, além de ter o potencial de mostrar todas as configurações do módulo através de simples requisições ARP. Apesar

Tabela 1. Vulnerabilidades encontradas no módulo EM1206 da Tibbo.

Camada	Problemas Encontrados
Ethernet	Etherleak
IP	*Flooding*
UDP	*Flooding*
TCP	*Flooding*, Aleatoriedade Fraca do Número Inicial de Sequência e *Reset Spoofing*
HTTP	Autocompletar Senha, *Flooding*, Roubo de *Cookie* e Transmissão de Senha Não Criptografada

de o teste ser de fácil execução, é muito difícil obter algum dado relevante através das requisições ARP, visto que os dados são obtidos aleatoriamente do *buffer* de rede, porém há sempre a possibilidade de o atacante obter algum dado relevante com poucas tentativas.

Também foi descoberto que o EM1206 é vulnerável a ataques *flooding*. Por um lado, os recursos em qualquer sistema são finitos, portanto todos os sistemas são vulneráveis a ataques *flooding*, quanto mais sistemas embarcados. Por outro lado este dispositivo é normalmente utilizado embarcado em circuitos eletrônicos, o que faz com que na maioria das vezes ele esteja em máquinas fechadas, onde pode ser impossível desligar e ligar apenas o EM1206, sendo necessário reiniciar o sistema inteiro. É importante que o administrador de rede leve isto em conta na hora de implementar o sistema.

Já na vulnerabilidade de aleatoriedade fraca do número inicial de sequência, vemos que o resultado errado fornecido pelo Nmap e a possibilidade de o Nessus gerar falsos positivos serve de alerta quanto à confiança cega nos resultados de ferramentas de testes, principalmente aquelas tidas como mais confiáveis. É importante antes de afirmar que uma vulnerabilidade existe ou não, realizar diversos testes com diferentes ferramentas.

Por fim, o uso do DS Manager mostrou-se a mais crítica de todas as vulnerabilidades encontradas. Este *software* é o primeiro a ser utilizado, pois é através dele que as primeiras configurações no EM1206 são feitas. O problema deste *software* é que ele envia todos os dados em *broadcast* nas camadas de rede e de enlace. Sendo assim, todos os *hosts* da rede recebem estes dados, inclusive a senha do módulo.

6. Conclusão

Dispositivos embarcados como conversores serial-*Ethernet* e microcontroladores são bastante disseminados em sistemas de controle industrial modernos, especialmente em sistemas de controle distribuídos. A crescente dependência desse tipo de sistema para o controle de infraestruturas críticas e para o controle dos mais variados tipos de processos industriais torna os componentes com acesso à rede alvos preferenciais de indivíduos ou organizações que queiram comprometer o bom funcionamento dessas infraestruturas e processos. Por conta disso, é importante que esses dispositivos operem de forma correta mesmo quando sujeitos a ataques. Com isso em mente, o objetivo deste trabalho foi avaliar a segurança contra ataques de rede de um dispositivo deste tipo, o módulo embarcado EM1206 da Tibbo.

Para o desenvolvimento deste trabalho foi aplicada uma técnica conhecida como Injeção de Faltas e a metodologia usada para garantir que um amplo espectro de faltas

fosse injetado foi baseada na proposta de Pothamsetty e Balinsky (2003), que tem por foco testar a robustez de implementações da pilha TCP/IP. Esta metodologia adotada primeiramente propõe que sejam executados testes *fuzzing*, que foram aplicados nas camadas *Ethernet*, UDP, TCP, IP e HTTP através das ferramentas ISIC e BED, porém não foram descobertas vulnerabilidades. Apesar disso, o fluxo de dados gerado pelas mesmas causou um ataque *flooding*.

Uma dificuldade encontrada durante o desenvolvimento deste trabalho foi achar materiais sobre testes de vulnerabilidades de rede especificamente em dispositivos embarcados e para contornar este problema, foi utilizado o *scanner* de vulnerabilidades Nessus que encontrou as seguintes vulnerabilidades: Etherleak, Aleatoriedade fraca do ISN, *Reset Spoofing*, Autocompletar Senha e Transmissão de Senha Não Criptografada. Além destas, através da interpretação do tráfego do EM1206 pelos autores deste trabalho, foi detectada a vulnerabilidade quanto ao Roubo de *Cookies*. Estas duas técnicas podem ser consideradas como uma contribuição à proposta de Pothamsetty e Balinsky.

As vulnerabilidades descritas neste trabalho podem ser utilizadas para falsificar a identidade de usuários e revelar informações confidenciais. Além disso, mostrou-se possível causar a negação do serviço. Estes problemas violam todas as propriedades de segurança, a saber, confidencialidade, integridade e disponibilidade [ISO/IEC 2008]. Levando em conta que o EM1206 pode estar inserido em sistemas críticos, pode-se dizer que o nível de segurança do módulo EM1206 é baixo. Contudo vale ressaltar que o microcontrolador EM1206 mostrou-se robusto diante dos ataques *fuzzing* executados.

Este tipo de análise voltada a dispositivos com poder computacional reduzido, normalmente usados em aplicações dedicadas e específicas, pode ser executada em outros equipamentos do mesmo gênero que possuem comunicação em rede, para que os resultados possam ser comparados e, a partir destes, novas metodologias de testes sejam criadas. Alguns exemplos são os outros módulos embarcados fabricados pela Tibbo, o Raspberry Pi[2], o BeagleBone[3] e o microcontrolador Arduino[4], que possui comunicação em rede através do uso de *shields* específicos.

Outra sugestão é a criação de uma metodologia para ataques *fuzzing*. A suite de ferramentas ISIC utilizada neste trabalho aplica *fuzzing* aleatoriamente nos pacotes de rede enviados. A criação de uma metodologia quanto ao uso de *fuzzing* permite que este tipo de teste possa ser melhor guiado e mais seletivo.

Além destes, poderiam ser criadas métricas para medir o quão robusto é um dispositivo deste gênero diante de ataques *flooding*. Estas informações podem ser úteis para que os responsáveis pela infraestrutura de rede onde estes equipamentos estão inseridos possam tomar precauções, com o objetivo de minimizar o impacto deste tipo de ataque.

Referências

Anderson, H. (2003). Introduction to Nessus. `http://goo.gl/9lme1u`.

Antunes, J. a., Neves, N., Neves, R., Correia, M., and Veríssimo, P. (2005). *Diagnóstico de Vulnerabilidades Através da Injeção de Ataques*. Actas da 1ªConferência Nacional Sobre Segurança Informática nas Organizações, Covilhã, Portugal.

[2]http://www.raspberrypi.org/

[3]http://www.beagleboard.org/Products/BeagleBone

[4]http://www.arduino.cc/

Arkin, O. and Anderson, J. (2003). Etherleak: Ethernet Frame Padding Information Leakage. http://goo.gl/6osJgP.

Arlat, J., Aguera, M., Amat, L., Crouzet, Y., Fabre, J., Laprie, J., Martins, E., and Powell, D. (1990). Fault Injection for Dependability Validation: A Methodology and Some Applications. *IEEE Transactions on Software Engineering*, 16(2):166–182.

Avizienis, A., Laprie, J., Randell, B., and Landwehr, C. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE TDSC*, 1(1):11–33.

Banks, G., Cova, M., Felmetsger, V., Almeroth, K., Kemmerer, R., and Vigna, G. (2006). *SNOOZE: Toward a Stateful NetwOrk prOtocol fuzZEr*. *Information Security*, LNCS 4176, p. 343-358.

Biondi, P. (2007). Scapy. http://goo.gl/8Hs7Sh.

Brudna, C. (2000). *Desenvolvimento de Sistemas de Automação Industrial Baseados em Objetos Distribuídos e no Barramento CAN*. Dissertação (Mestrado em Engenharia Elétrica) - Escola de Engenharia, Universidade Federal do Rio Grande do Sul.

Clark, J. and Pradhan, D. (1995). Fault injection: A method for validating computer-system dependability. *IEEE Computer*, 28(6):47–56.

Combs, G. (s.n.t.). About Wireshark. http://goo.gl/jnRRmW.

Damaye, S. (2012). BED. http://goo.gl/RhgdgF.

Deraison, R. (2002). Raptor Weak ISN. http://goo.gl/5vE55y.

Gont, F. (2012). RFC 6528: Defending against sequence number attacks.

Gu, S., Song, Y., Zhao, X., and Li, W. (2011). Fuzzing test data generation based on message matrix perturbation with keyword reference. In *Military Communications Conference, (MILCOM 2011)*. IEEE.

Hart, D. (2004). *An Approach to Vulnerability Assessment for Navy Supervisory Control And Data Acquisition (SCADA) Systems*. Storming Media.

Hsueh, M., Tsai, T., and Iyer, R. (1997). Fault Injection Techniques and Tools. *IEEE Computer*, 30(4):75–82.

ISO/IEC, I. L. (2008). ISO/IEC 27002:2005 information technology – security techniques – code of practice for information security management.

Kurose, J. F. and Ross, K. W. (2006). *Redes de Computadores e a Internet, Uma abordagem top-down*. Pearson Education do Brasil, São Paulo, SP, Brasil.

Lyon, G. F. (s.n.t.). Nmap Network Scanning. http://goo.gl/jPdt9J.

Melton, R., Fletcher, T., and Earley, M. (2004). *System Protection Profile - Industrial Control Systems*. National Institute of Standards and Technology.

Oehlert, P. (2005). Violating assumptions with fuzzing. *IEEE Security & Privacy, 3(2):58-62*.

Postel, J. (1988). RFC 1042. A Standard For The Transmission Of IP Datagrams Over IEEE 802 Networks. http://goo.gl/KlwkgA.

Pothamsetty, V. and Balinsky, A. (2003). A structured and practical methodology for security evaluation of a IP based stack (version 0.2). http://goo.gl/Y5QMAf.

Ralston, P., Graham, J., and Hieb, J. (2007). Cyber security risk assessment for SCADA and DCS networks. *ISA transactions*, 46(4):583–594.

Stouffer, K., Falco, J., and Scarfone, K. (2007). Guide to industrial control systems (ICS) security. *NIST Special Publication 800-82*.

Tibbo (2013). Telnet TCP Programming. http://goo.gl/IvF2et.

Xiao, S. and Frantzen, M. (s.n.t.). ISIC – IP stack integrity checker. http://goo.gl/hX5K8R.

Um Mecanismo de Segurança para o Protocolo HTR

Gregório Patriota[1], Eduardo Feitosa[2], Djamel Sadok[1]

[1]CIn/UFPE, Recife, Brasil

{gregorio, jamel}@gprt.ufpe.br, efeitosa@icomp.ufam.edu.br

Abstract. *Data security has become an essential requirement for any system in the network to adhoc networks this would be no different. Among the range of existing adhoc protocols no focus on protecting the data that travels. The Heterogeneous Technologies Routing (HTR) is a protocol developed to provides an efficient data routing with the lowest energy consumption, but their data travels in the clear between the elements that make up the network. This work aims to evaluate the impact of a security system on Mobile adhoc network HTR.*

Resumo. *Segurança de dados tornou-se um requisito essencial para todo e qualquer sistema em rede, para as redes adhoc isto não seria diferente. Entre a gama de protocolos para adhoc existentes não há foco na proteção dos dados que trafegam. O Heterogeneous Technologies Routing (HTR) foi um protocolo desenvolvido que provê um roteamento eficiente de dados com o menor consumo energético, porém seus dados trafegam em claro entre os elementos que compõem a rede. Este trabalho tem como foco avaliar o impacto causado por um sistema de segurança sobre a rede adhoc móvel HTR.*

1. Introdução

Recentemente foi elaborado um novo *framework* de roteamento para interligar dispositivos em um ambiente de redes adhoc móveis (MANET), chamado HTR (*Heterogeneous Technologies Routing*) [Souto et al. 2012]. Ele cria uma MANET heterogênea fechada capaz de lidar com mudanças rápidas e imprevisíveis da topologia da rede. Essas MANETs podem ser configuradas rapidamente em diversos ambientes e podem ser compostas de diferentes tecnologias de comunicação, como Bluetooth, Wi-Fi, 6LoWPAN, Zigbee e Ethernet. HTR também fornece suporte de auto-organização para inicializar seus nós, requerendo o mínimo de interação humana.

Por ter sido projetada para cenários de emergência, uma rede HTR parte do pressuposto que todos os nós que a compõem são confiáveis, ou seja, não há nós que praticam ações maliciosas. Desta forma, todas as informações de dados e de controle trafegam em texto claro, sem controles de integridade e autenticidade da informação. Em outras palavras, as preocupações de segurança foram deixadas de lado. Porém, tal situação não condiz com a realidade. A ausência de uma estrutura física permite que os canais de comunicações fiquem abertos, possibilitando que um elemento não autêntico da rede possa interceptar o sinal e capturar os dados. Outras questões como ausência de um elemento central coordenador, largura de banda e estabilidade limitam os mecanismos de criptografias que poderiam ser utilizados para proteger o canal aberto de comunicação entre os nós.

Assim, este trabalho desenvolve uma versão segura para o *framework* HTR, chamada S-HTR. A solução proposta tem como objetivo garantir os pilares da integridade e autenticidade para o ambiente HTR. A ideia base é utilizar um esquema de HMAC (*Hash-based MAC*) para prover uma interface entre funções de criptografia.

O restante do artigo é organizado como segue: na Seção 2 são apresentados os conceitos básicos relacionados ao tema. A Seção 3 descreve sucintamente as mudanças na arquitetura HTR para a criação do S-HTR. A Seção 4 explica todo o protocolo experimental deste trabalho. Na Seção 5 são apresentados as análises dos resultados obtidos via ambiente de simulação, bem com uma discussão sobre a adequação do HTR a certos ataques. Por fim, na Seção 6 estão expostas as conclusões obtidas, as dificuldades encontradas e as possibilidades de trabalhos futuros.

2. Conceitos Básicos

Essa seção apresenta os conceitos básicos necessários para o melhor entendimento deste trabalho.

2.1. HTR

Heterogenous Technologies Routing (HTR) é um protocolo de roteamento próativo focado em apoiar heterogeneidade em termos de tecnologia de comunicação [Souto et al. 2012]. Baseado no protocolo OLSR [Clausen and Jacquet 2003], o HTR usa as mensagens de *Topology Control* (TC) e *HELLO* para disseminar informação sobre o estado do enlace (*link state*) e vizinhança (*neighbor*). Além disso, herda *Multipoint Relays* (MPRs), técnica para evitar o controle de inundação de mensagem escolhendo alguns nós para serem responsáveis pela disseminação de mensagens de controle. Contudo, o HTR alcança melhores tempos de convergência [Lima et al. 2013].

O roteamento do HTR utiliza o algoritmo de Dijkstra para executar a computação do caminho [Souto et al. 2012], mas também utiliza uma abordagem *multipath* [Lima et al. 2014] para encontrar melhores caminhos no processo de roteamento. Uma das diferenças entre o HTR e outros protocolos de redes adhoc reside no fato dele ter como foco a manutenção da rede em operação pelo maior tempo possível. Isso é alcançado pelo uso da métrica HTRScore, definida usando fatores como estado do enlace e eficiência energética [Souto et al. 2012]. Esta métrica é usada para melhorar o cálculo de rota e também é aplicável ao conjunto de cálculos do MPR. HTRScore também introduz a probabilidade de perda de pacotes, um parâmetro que avalia a estabilidade do link, uma otimização que beneficia caminhos com alta taxa de emissão de pacotes.

2.2. Ataques a Redes Adhoc

Os ataques a redes adhoc visam, em geral, a tabela de roteamento dos elementos que a compõem. Como forma de facilitar o entendimento do problema e, mais a frente, da solução proposta, os principais ataques a redes adhoc encontrados na literatura [Agrawal et al. 2011] são descritos na Tabela 1.

2.3. Função *hash*

Uma função *hash* [Stallings 2007] mapeia uma mensagem de tamanho variável em um resumo de mensagem com tamanho fixo. Também conhecida como síntese de

Tabela 1. Principais Ataques a Redes AdHoc

Ataque	Descrição
Inundação (*Flooding Attack*)	Visa sobrecarregar os nós autênticos com a criação de rotas falsas com destino para nós não existentes. É classificado com um tipo de ataque de negação de serviço onde os protocolos proátivos são os mais susceptíveis
Sleep Depravation	Com o foco no consumo energético de um determinado nó, neste ataque o atacante faz consecutivas requisições de rota sobrecarregando um elemento da rede. O nó atacado irá desperdiçar a sua energia processando as requisições do nó malicioso
Impersonation Attack	Um nó malicioso finge ser um nó legítimo da rede. Dessa forma, junta-se a rede e envia informações falsas de rotas, mascarando os nós autênticos
Black Hole Attack	É um ataque de negação de serviço em que o atacante injeta rotas falsas na rede, afirmando ter o menor caminho para um dado destino, atraindo todo o tráfego para ele. Desta forma, o atacante pode interceptar os pacotes e fazer mau uso deles
Node Isolation Attack	O atacante tem como meta isolar um determinado nó dentro da rede, evitando que rotas sejam construídas para este nó destino.
Routing Table Poisoning Attack	O atacante gera e envia tráfego falso ou altera o conteúdo de uma mensagem autentica, criando entradas falsas nas tabelas de roteamento
Wormhole Attack	Um atacante replica os dados que trafegam em um determinado ponto da rede em outro ponto. Ou seja, os dados são inseridos em um túnel que liga dois pontos distintos da rede
Location Disclosure Attack	A partir de uma análise de tráfego, o atacante pode descobrir a localização de um nó dentro da rede, podendo descobrir também a topologia da rede.
Rushing Attack	Aproveitando o esquema de requisição de rotas nas redes adhoc, o atacante passa a responder o mais rápido essas requisições, de forma a garantir que todas as rotas tenham como elemento intermediário o nó malicioso. As requisições que são respondidas com atraso são descartadas pelos nós
Blackmail Attack	Ataque contra protocolos que implementam um sistema seguro baseado em *blacklist*. O atacante cria a própria *blacklist* e propaga dentro da rede. Nela o atacante inseri elementos autênticos e retira nós maliciosos.
Snare Attack	Trata-se de um ataque físico, onde um elemento da rede é capturado. O atacante pode usar o dispositivo, se passando por um indivíduo autentico da rede, interceptando o tráfego.
The Invisible Node Attack	É um ataque passivo, onde o nó malicioso participa da ações da rede sem revelar sua identidade, com o intuito de apenas coletar as informações que trafegam na rede

mensagem ou valor de *hash*, a função *hash* é uma função unidirecional, pois a partir do resumo gerado não é possível recuperar a mensagem original. O código de *hash* é uma função que utiliza todos os bits da mensagem e oferece a capacidade de detecção de erros. Qualquer mudança de bit ou bits na mensagem resulta em uma mudança no código *hash*.

Um valor de *hash h* é gerado por uma função *H* na forma *h=H(M)*, onde *M* é uma mensagem de tamanho variável e *H(M)* é o valor de *hash* de comprimento fixo. A finalidade de uma função *hash* é produzir uma 'impressão digital' de um arquivo, mensagem ou qualquer outro conjunto de dados.

2.4. MAC

O Código de Autenticação de Mensagem (*Message Authentication Code* - MAC)[Stallings 2007], também conhecido como soma de verificação criptográfica (*checksum*), é uma técnica de autenticação que envolve o uso de uma chave secreta para gerar um bloco de dados de tamanho fixo. Este bloco é anexado à mensagem na transmissão no momento em que a mensagem é conhecida como sendo correta. Na recepção, a mensagem é autenticada a partir de um novo MAC gerado e comparado. Para geração do novo código, tanto o receptor quanto o elemento que transmite a mensagem, devem estar de posse da chave secreta.

Com o uso de MAC é possível garantir que: (i) a mensagem recebida não foi alterada; (ii) a mensagem recebida é do emissor declarado; (iii) se a mensagem inclui um número de sequência, então o receptor poderá ter a certeza que a mensagem está na devida sequência. O algoritmo de MAC é uma função irreversível, ou seja, não é possível a partir do código gerado obter a mensagem original.

2.5. HMAC

Uma vez que algumas funções *Hash* não foram projetadas para uso com o MAC (Código de Autenticação de Mensagem - *Message Authentication Code*), pois não dependem de uma chave secreta, várias propostas surgiram para incorporar uma chave simétrica em algoritmo de *Hash*. A técnica que obteve maior suporte foi *Hash-based MAC* (HMAC) [Stallings 2007]. Publicado na RFC2104 [Krawczyk et al. 1997] e escolhido como MAC de implementação obrigatória para segurança IP, O HMAC é usada em protocolos da Internet, como SSL.

O HMAC trata a função *hash* como uma 'caixa-preta', ou seja, como um módulo independente onde o esquema HMAC não precisa ter conhecimento de seu funcionamento. Isso o torna adaptável a qualquer nova função *hash* que possa vir a surgir e facilita a troca de uma função *hash* por outra, sem necessidade de uma reimplementação. A força da segurança do HMAC está diretamente associada ao tamanho da chave simétrica utilizada e a segurança do algoritmo de *Hash* utilizado. Uma chave pequena pode comprometer a segurança do esquema, mas uma chave grande não aumentará mais o nível de segurança. A relação de tamanho ideal para a chave é expressa em [Krawczyk et al. 1997] .

2.6. Protocolos Seguros

Embora existam soluções e protocolos seguros, como SAODV, SDSDV, SOLSR, entre outros (todos descritos em [Abusalah et al. 2008]), este trabalho não tem o propósito de fazer uma comparação direta entre o S-HTR e as outras soluções. Por isso elas não serão exploradas e/ou detalhadas neste trabalho.

3. Arquitetura

Para alcançar o objetivo proposto, prover um ambiente seguro para o tráfego das mensagens de controle do protocolo HTR, foi utilizado um esquema de HMAC. Como explicado na Seção 2.1, o *framework* HTR usa dois tipos de mensagens de controle: HELLO e TC. Assim, para implementar a versão segura do HTR, o cabeçalho padrão do protocolo foi alterado (Figura 1).

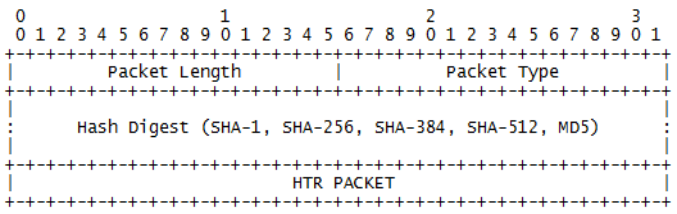


Figura 1. Cabeçalho da mensagem de controle do S-HTR

Os campos *Packet Length* e *Packet Type*, ambos de 2 bytes cada um, representam, respectivamente, o tamanho total da mensagem e o tipo da mensagem (HELLO ou TC). Para o S-HTR, independente da mensagem encapsulada, os campos dessa mensagem serão usados como entrada para a função *hash*. Ou seja, o resumo gerado será composto por todos os campos da mensagem encapsulada. O tamanho do campo *HashDigest* varia conforme a função *hash* utilizada, podendo ser de 128, 160, 256, 384 e 512 bits (conforme disposto na Tabela 2). O resumo é encriptado por um protocolo de criptografia simétrica AES, que utiliza uma chave

previamente trocada entre os participantes para fornecer o código de autenticação da mensagem baseado em *hash*. O esquema HMAC para as mensagens do HTR é ilustrado na Figura 2.

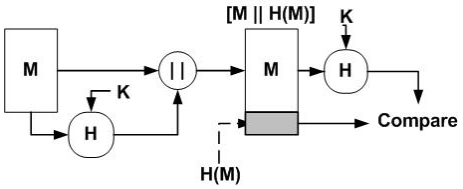


Figura 2. Estrutura do HMAC no S-HTR

No processo de transmissão, a partir de uma mensagem M e uma chave simétrica K é gerado o código HMAC da mensagem M pela função *hash* H e a chave simétrica K. O código é concatenado com a mensagem e então propagado para o devido destino. Na recepção da mensagem, a partir da mensagem recebida M e de uma chave simétrica K, a mesma utilizada na etapa de transmissão, é então gerado um novo código HMAC da mensagem. O novo código é comparado com o código recebido na mensagem. Esse processo de comparação é responsável por verificar e garantir a integridade e autenticidade da mensagem.

4. Protocolo Experimental

A solução proposta foi desenvolvida em ambiente simulado, através do simulador ns-3 [NS-3 Consortium 2014] e da biblioteca Crypto++ [Boost Software License 1.0 2013]. As simulações foram executadas em computadores com 8 processadores quad-core de 3.4GHz, com 8Gb de memória e 1Tb em disco. A versão do sistema operacional foi o Debian GNU/Linux 7.1 Wheezy.

4.1. Cenário

Os elementos que compõem a rede foram alinhados com uma distância de 30 metros separando uns dos outros. O protocolo utilizado na camada de transporte foi o UDP, com taxa de dados de 300, 600 e 1000kbps. Os nós são estáticos na rede durante todo o processo de simulação. Os experimentos foram executados 400 vezes para cada um dos cenários com duração máxima de 1800 segundos. Cada cenário possuí uma densidade de nós e um protocolo. A densidade foi variada entre 25, 49, 64 e 81 nós na rede, valores estes escolhidos pelo autor. Já entre os protocolos utilizados, foi permutado entre HTR, S-HTR(MD5), S-HTR(SHA-1), S-HTR(SHA-128), S-HTR(SHA-256), S-HTR(SHA-384), S-HTR(SHA-512).

4.2. Métricas

Foram empregadas as três (3) mesmas métricas de avaliação utilizadas em [Junior 2013, Lima et al. 2013]. São elas:

1. **Tempo de Convergência**: é o tempo que a rede leva para que a tabela de roteamento de cada elemento esteja preenchida com rotas para todos os nós restantes que compõem a rede.
2. **Sobrecarga das Mensagens**: é definida como o fluxo médio de dados de controle ao longo da simulação. A sobrecarga do novo cabeçalho é diretamente proporcional a função *Hash* utilizada (vide Tabela 2).
3. **Perda de Pacotes**: é o número de pacotes transmitidos dentro da rede que não são recebidos.

Tabela 2. Tamanho dos resumos gerados pelas respectivas funções *Hash*

Algoritmo	Tamanho da Saída (bits)
MD5	128
SHA-1	160
SHA-3-256	256
SHA-3-384	384
SHA-3-512	512

5. Avaliação de Resultados

Esta seção apresenta e discute os resultados das avaliações.

5.1. Resultados

5.1.1. Tempo de Convergência

A Figura 3 apresenta os tempos de convergência nos diferentes cenários. Para o cenário de menor densidade(a), 25 nós, percebe-se que a diferença entre o HTR e os S-HTR é mínima, com variação menores do que 2%. Valor este que, além de estar dentro da margem de erro, testes estatísticos determinaram que os valores de suas medianas não diferem em localidade. Contudo, conforme a densidade da rede cresce o impacto torna-se notório. Para o cenário com 49 nós (b), o impacto do S-HTR em relação ao HTR chega 6% (em média). Para os cenários com 64 nós (c) e 81 nós (d), o tempo de convergência dos protocolos com o mecanismo de segurança são visivelmente maiores que o protocolo base. Tal fato se deve ao tempo de processamento que cada função *Hash* demanda e conforme a rede vai se expandindo, o impacto vai aumentando.

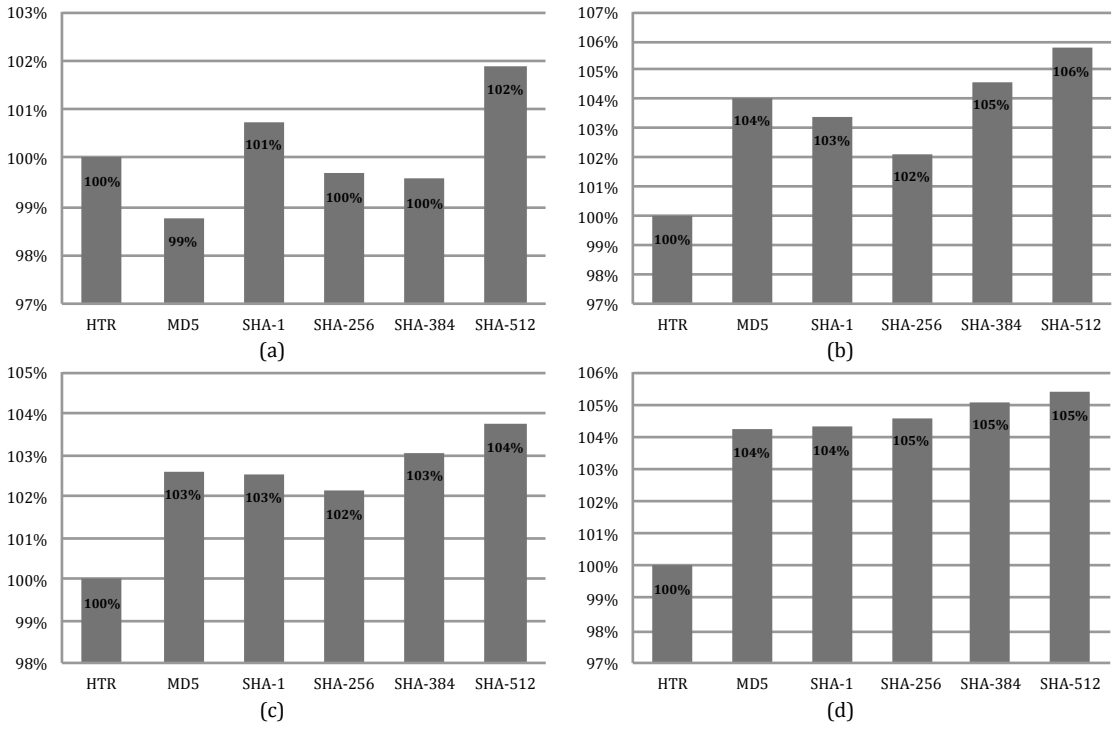


Figura 3. Comparação entre os tempos de convergência para cenários com: (a) 25 nós, (b) 49 nós, (c) 64 nós, e (d) 81 nós.

5.1.2. Sobrecarga das Mensagens

A Figura 4 apresenta a sobrecarga das mensagens nos diferentes cenários. Como mencionado na Seção 4.2, a carga depende da função *Hash*. Em outras palavras, o novo cabeçalho de segurança insere uma sobrecarga nas mensagens.

Isto é facilmente percebido no cenário com 25 nós (a), onde a sobrecarga é de quase 100% na comparação do HTR com o protocolo com esquemas criptográficos. Para os cenários maiores (b), (c) e (d), esta proporção na sobrecarga se mantém. Em média, a cada função *Hash* utilizada no protocolo, a quantidade de dados no cabeçalho da mensagem dobra em relação ao HTR. Contudo, também pode-se observar que apesar do crescimento em relação ao HTR, quando comparados entre si, a sobrecarga da mensagem mantém uma proporção igual. Isso significa que, independentemente do protocolo seguro utilizado, os custos de banda são muito próximos, com pouca variação.

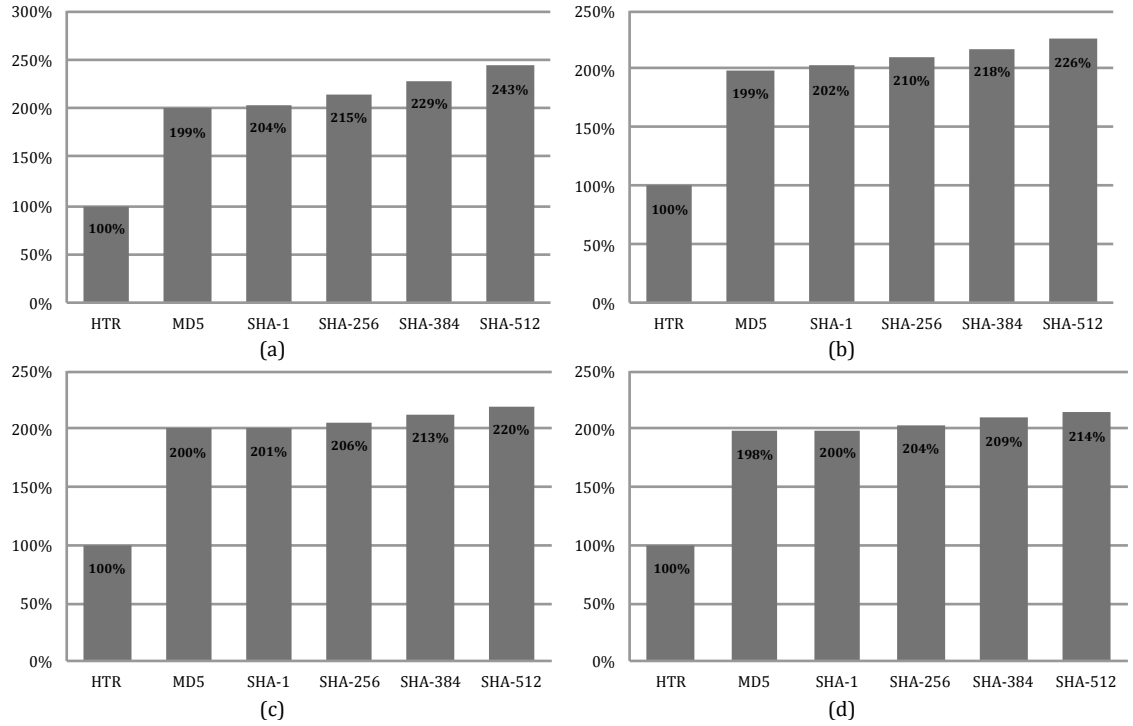


Figura 4. Comparação entre as cargas de dados de controle para cenários com: (a) 25 nós, (b) 49 nós, (c) 64 nós, e (d) 81 nós.

5.1.3. Perda de Pacotes

No caso da métrica de perda de pacotes, todos os protocolos seguros apresentaram um aumento na taxa de perda de pacotes, como pode ser visto na Figura 5. No cenário de 25 nós (a), os pacotes perdidos chegam a ser de 50% quando comparados com o HTR. Conforme a densidade da rede aumenta, o impacto dos protocolos seguros sobre a perda de pacotes aumenta também. Porém, conforme observado na Figura 5 (b), (c) e (d), o impacto causado pelos protocolos seguros, quando comparados entre si, tendem a manter um nível médio igual, ou seja, não apresentam

grandes variações. A perda de pacotes está diretamente associado à qualidade do canal que é definido pelo modelo de propagação.

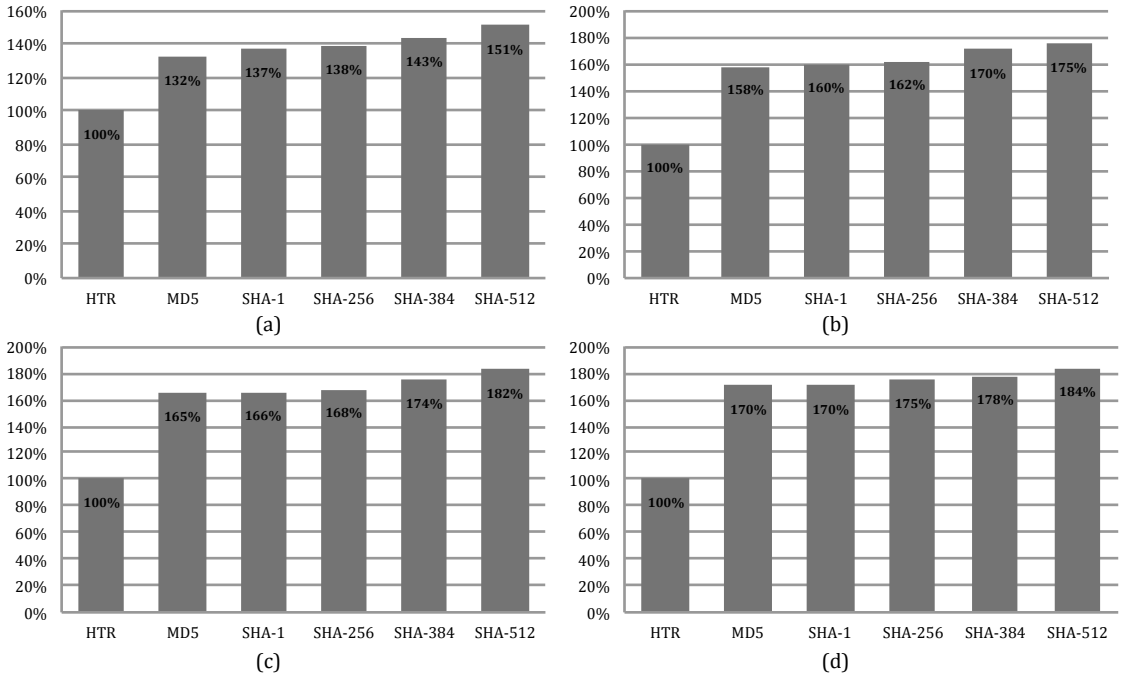


Figura 5. Comparação entre a perda de pacotes de controle para cenários com: (a) 25 nós, (b) 49 nós, (c) 64 nós, e (d) 81 nós.

5.2. Discussão

Ao analisar as três métricas, é notório que existe um custo associado à geração de um resumo para cada função *Hash*, o que influência no tempo de convergência da rede, na sobrecarga das mensagens e na perda de pacotes. Sobre o Tempo de Convergência cada função de Hash influncia diretamente, porém para o ambiente HTR, os dados que são a entrada para as funções de Hash são os dados das mensagens de controle, a carga desses dados é muito pequena, na ordem de dezenas de *bytes*. Sendo assim o tempo de convergência dos protocolos com seus respectivos esquemas de Hash não diferiu tanto do protocolo HTR sem criptografia.

Para todos os protocolos seguros, a sobrecarga da rede é maior que o dobro em relação ao protocolo HTR sem criptografia. Mas quando comparamos o custo entre os protocolos seguros, o custo de banda é muito próximo com pouca variação de pontos percentuais. Isto é devido aos resumos inseridos que, conforme a Tabela 2, entre eles há apenas uma pequena diferença de tamanho. Quanto as perdas de pacotes causadas pelos protocolos seguros, observamos que para redes com uma alta densidade de nós, a perda de pacotes chega a quase dobrar, porém não há uma grande variação de impacto quando comparamos os protocolos seguros entre si.

Contudo, ao se considerar o aspecto de segurança, um ambiente com o S-HTR é capaz de mitigar vários dos ataques descritos na Seção 2.2. Embora uma análise em ambiente real tenha sido realizada, uma rápida análise teórica permite (Tabela 3) apontar em quais desses ataques o S-HTR consegue atuar.

Tabela 3. Status dos Ataques em um ambiente com o S-HTR

Ataque	Status	Explicação
Inundação (*Flooding Attack*)	Mitiga	Apenas nós autênticos na rede irão ter posse da chave simétrica. No caso dos elementos que não possuem a chave simétrica, não conseguirão gerar pacotes com o resumo *Hash*.
Sleep Depravation	Não se enquadra	Este tipo de ataque tem impacto em protocolos reativos, onde as rotas são montadas no instante em que são requisitados. O S-HTR ó um protocolo próativo onde os destinos são armazenados em tabelas, logo não há requisição de rota.
Impersonation Attack	Não Mitiga	ó necessário um esquema de troca periódica de chaves entre os nós autênticos, garantindo que apenas os nós autênticos terão acesso as chaves simétricas e assimétricas.
Black Hole Attack	Mitiga	A escolha do melhor caminho no HTR ó calculada através do algoritmo de Dijkstra, onde os pesos dos enlaces são ponderados pelo HTRScore.
Node Isolation Attack	Não Mitiga	ó um ataque criado especificamente contra o OLSR, logo, como o HTR herda várias características do OLSR ele também é vulnerável a este ataque. O S-HTR não implementa mecanismos para corrigir este tipo de ataque.
Routing Table Poisoning Attack	Mitiga	O uso da função *Hash* com a chave simétrica protege o conteúdo das mensagens de controle, garantindo assim a integridade dos dados. Logo não é possível que um nó não autêntico altere ou falsifique os dados das mensagens de controle.
Wormhole Attack	Não Mitiga	Não foi encontrada uma solução tangível para resolver este tipo de ataque.
Location Disclosure Attack	Não Mitiga	Para evitar este ataque ó preciso que todo o conteúdo seja criptografado. O S-HTR usa o esquema do HMAC que criptografa apenas o resumo *Hash* gerado.
Rushing Attack	Mitiga	No *framework* HTR, a seleção de melhor rota é atribuída ao cálculo do MPR. O S-HTR mantém esta funcionalidade.
Blackmail Attack	Não se enquadra	Nem HTR e nem o S-HTR fazem uso de mecanismos de segurança de *Blacklist*, logo este ataque não se enquadra.
Snare Attack	Não Mitiga	Hó necessidade de autenticação biométrica para evitar este ataque.
The Invisible Node Attack	Não Mitiga	Ataques do tipo passivo, onde o atacante não tem influência alguma sobre a rede, são difíceis de serem detectados.

6. Conclusão

Prover segurança é um requisito imprescindível para qualquer sistema e rede atualmente. Para uma rede adhoc esse requisito tem maior importância devido à alta vulnerabilidade e mobilidade inerente a sua arquitetura. Entretanto, por serem construídas por elementos geralmente com restrições de recursos, o uso de mecanismos seguros também pode significar aumentos no custo de processamento, do uso da banda e de memória, impossibilitando ou inviabilizando sua utilização.

Neste trabalho foi feita uma análise do impacto de um mecanismo de segurança sobre a rede adhoc implementada usando o protocolo HTR. Como contribuição, este trabalho foi capaz de gerar argumentos que permitem a escolha do melhor esquema de segurança para o protocolo de redes adhoc, o HTR. A questão de melhores garantias de segurança e de menor impacto sobre a rede foram avaliadas, demonstrando o impacto causado pelas funções: MD5, SHA-1, SHA-256, SHA-384 e SHA-512.

Embora tenha as funções *Hash* tenham gerado impacto sobre a rede (tempo de convergência, sobrecarga das mensagens e até perda de pacotes), pode-se afirmar que: (1) funções SHA-384 e SHA-512 irão sobrecarregar a rede de forma desnecessária; (2) função MD5 não é confiável [Turner and Chen 2011]; e (3) função SHA-1 apresentou significativamente o impacto menor e SHA-256 o máximo impacto suportado.

Desta forma, os resultados dos testes e a arquitetura do protocolo HTR nos levam a concluir que o esquema de HMAC com uso da função *Hash* SHA-1 trará menor impacto e garantirá, de forma eficiente, integridade e autenticidade das mensagens de controle do *framework*, já que o tempo de vida da mensagem de controle [Souto et al. 2012] é muito inferior ao tempo necessário para a criptoanálise do SHA-1.

6.1. Dificuldades Encontradas

Entre as dificuldades encontradas, destacam-se:

- Falta da implementação de um modelo de consumo energético próprio para cada função de *Hash* utilizada no ns-3, o que impossibilitou que esta métrica pudesse ser coletada.
- A segunda e maior dificuldade encontrada reside no fato da biblioteca *Crypto++* possuir uma pequena falha de gerenciamento da memória. Tal falha ocasiona um vazamento de memória, conhecido como *memory leak*. Esse vazamento era evidente em cenários de alta densidade de nós, o que impossibilitou a simulação de cenários maiores.

6.2. Trabalhos Futuros

Como propostas de trabalhos futuros: (i) criação de um esquema seguro de distribuição de chaves simétricas em ambiente inseguro; (ii) extensão para o protocolo para dar suporte a CMAC; (iii) criação de modelos de energia no ns-3 para cada função *Hash*.

Referências

Abusalah, L., Khokhar, A., and Guizani, M. (2008). A survey of secure mobile ad hoc routing protocols. *Commun. Surveys Tuts.*, 10(4):78–93.

Agrawal, S., Jain, S., and Sharma, S. (2011). A Survey of Routing Attacks and Security Measures in Mobile Ad-Hoc Networks. 3(1):41–48.

Boost Software License 1.0 (2013). Crypto++® library 5.6.2. `http://www.cryptopp.com/`.

Clausen, T. and Jacquet, P. (2003). Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental).

Junior, J. (2013). An energy-aware multipath routing extension for heterogeneous ad hoc networks. Master's thesis, Federal University of Pernambuco. Master Thesis.

Krawczyk, H., Bellare, M., and Canetti, R. (1997). HMAC: Keyed-Hashing for Message Authentication. RFC 2104 (Informational).

Lima, J., Rodrigues, T., Melo, R., Correia, G., Kelner, J., and Feitosa, E. (2013). On the tuning of wireless heterogeneous routing. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 3391–3396. IEEE.

Lima, Josias, J., Rodrigues, T., Melo, R., Correia, G., Sadok, D., Kelner, J., and Feitosa, E. (2014). A multipath extension for the heterogeneous technology routing protocol. In Guo, S., Lloret, J., Manzoni, P., and Ruehrup, S., editors, *Ad-hoc, Mobile, and Wireless Networks*, volume 8487 of *Lecture Notes in Computer Science*, pages 15–28. Springer International Publishing.

NS-3 Consortium (2014). ns-3. `https://www.nsnam.org`.

Souto, E., Aschoff, R., Junior, J. L., Melo, R., Sadok, D., and Kelner, J. (2012). Htr: A framework for interconnecting wireless heterogeneous devices. In *CCNC*, pages 645–649. IEEE.

Stallings, W. (2007). *Criptography and Networking Security*. Pearson Education, 4 edition.

Turner, S. and Chen, L. (2011). RFC 6151.

Uma análise do Impacto do Intervalo de Tempo de Captura do Acelerômetro na Biometria baseada em gestos em dispositivos móveis usando Android

Paulo Fernando de Magalhães Dreher[1], **Luciano Ignaczak**[1]

[1]Universidade do Vale do Rio do Sinos (Unisinos)
Bairro Cristo Rei – 93.022-000 – São Leopoldo – RS – Brasil

paulodreher@gmail.com, lignaczak@unisinos.br

Abstract. *The application of behavioural biometrics for authentication on mobile devices has been demonstrated in several studies, which show the feasibility of using gestures for authenticating users in a system. However, little or no attention has been paid to the way the movement is exercised and the period between capture points of a gesture. This article aims to analyse the security of different intervals of capture of an accelerometer applied to gestures on mobile devices. The analysis of the intervals was realized through an experiment, based on an application developed for Android platform, where two types of movement and three different capture intervals of the accelerometer were executed. The findings demonstrate that by reducing the capture interval of accelerometer, the safety of the use of biometrics by gestures increases. The results also highlight the importance of using complex movements to make the authentication process less susceptible to attacks.*

Resumo. *A aplicação de biometria comportamental para autenticação em dispositivos móveis vem sendo demonstrada em diversos trabalhos, os quais apresentam a viabilidade do uso de gestos para a autenticação de usuários em sistemas. Entretanto, pouca ou nenhuma atenção têm sido dispensada para a maneira como é exercido o movimento e o período de captura entre os pontos de um gesto. Este artigo tem como objetivo analisar a segurança de diferentes intervalos de captura das coordenadas de um acelerômetro aplicados à biometria por gestos em dispositivos móveis. A análise dos intervalos foi realizada através de um experimento, baseado em um aplicativo desenvolvido para a plataforma Android, onde foram executados dois tipos de movimentos considerando três diferentes intervalos de captura das coordenadas do acelerômetro. Os resultados do trabalho demonstram que a redução do intervalo de captura do acelerômetro aumenta a segurança do uso da biometria por gestos. Os resultados também evidenciam a importância de utilizar movimentos complexos para tornar o processo de autenticação menos suscetível a ataques.*

1. Introdução

O uso de dispositivos móveis, segundo o estudo realizado pela International Telecommunications Union (ITU), tem crescido constantemente e atingiu a marca de 6,8 bilhões de usuários [SANOU 2013]. Contudo, a autenticação em dispositivos móveis ainda precisa ser melhor analisada, pois segundo observam [BEN-ASHER et al. 2011, p. 466] "em

muitos casos o único mecanismo de segurança em telefones móveis é um Personal Identification Number (PIN)", sobre o qual ainda afirmam "sofrer de numerosos problemas de segurança e usabilidade". Um estudo nesse sentido, conduzido pelos autores [BONNEAU et al. 2012], utilizou os resultados obtidos inicialmente das análises de PINs de 4 dígitos em duas bases de dados, como comparativo para estimar o comportamento em relação a contas de banco. As análises demonstraram que, aproximadamente, 7% dos usuários utilizam um PIN associado a sua data de aniversário, e 61% desses usuários portavam informações sobre seus aniversários nas suas carteiras.

No sentido de contrapor tal cenário, diversos autores têm apresentado a biometria como novo método de autenticação por meio do qual propõem uma melhor forma de prover segurança no acesso às informações. A biometria, conforme explicam os autores [GUERRA-CASANOVA et al. 2012], pode ser segmentada de acordo com as características físicas pertencentes a um indivíduo, que mantém-se com o passar do tempo, e comportamental, relacionada à capacidade em executar algo de um modo único e singular.

Ainda conforme [GUERRA-CASANOVA et al. 2012, p.65], "um dos próximos passos na indústria da segurança é adaptar ou criar novas técnicas biométricas válidas para dispositivos móveis". Seguindo essa previsão, diversas propostas de técnicas biométricas envolvendo o uso de gestos vem sendo apresentadas, como as presentes nos trabalhos de [GUERRA-CASANOVA et al. 2012], [SAE-BAE et al. 2012] e [LUCA et al. 2012]. Contudo, em nenhum desses trabalhos foi estudado se o acelerômetro possui impacto na segurança de um sistema biométrico baseado em gestos.

Dessa forma, esse artigo tem como objetivo analisar a segurança de diferentes movimentos e intervalos de captura das coordenadas de um acelerômetro aplicados à biometria por gestos em dispositivos móveis. A análise de segurança será baseada em um experimento, onde serão executados dois tipos de movimentos considerando três diferentes intervalos de captura das coordenadas do acelerômetro.

Na seção 2 deste artigo são apresentados os trabalhos relacionados com o uso de biometria em dispositivos móveis. O experimento proposto é descrito na seção 3 e os seus resultados na seção 4. As considerações finais sobre o artigo são tratadas na seção 5.

2. Trabalhos Relacionados

Atualmente diversos trabalhos demonstram a viabilidade do uso da biometria por gestos, como [SAE-BAE et al. 2012], que apresentam um sistema de gestos multi-toque pelo qual o usuário registra os movimentos dos cinco dedos sob a tela de um IPad com o sistema operacional iOS 3.2 e, posteriormente, os compara com base em um algoritmo de reconhecimento. Os autores objetivam um valor próximo de 90% de acuracidade e, também, obter um alto grau de satisfação dos usuários, uma vez que consideram essencial a aceitabilidade do método. Bem por isso, classificam 22 possíveis gestos sob o dispositivo, onde são analisados os movimentos dos 5 dedos, como também, o posicionamento dos dedos polegar e indicador. Tais movimentos são mensurados pela distância percorrida no que os autores definem como eixos "x" e "y" através do algoritmo DTW [1]. Esses resultados são então avaliados por um classificador que determina se o usuário é legítimo

[1] Segundo [LUCA et al. 2012], Dynamic Time Warping é um algoritmo que compara uma sequência de tempo com outra a fim de encontrar similaridade elas. Ainda segundo o autor, este algoritmo é utilizado para reconhecimento de voz.

ou não. Os testes foram executados por 34 participantes, que ao final tiveram os seus resultados avaliados, considerando os níveis de acuracidade e aceitabilidade de determinados movimentos realizados. A conclusão do trabalho, segundo os autores, apesar de ter sido positiva, apresentaria melhores resultados se fossem inseridos outros parâmetros relacionados a gestos sob uma tela, como por exemplo, o nível de pressão.

No trabalho de [LUCA et al. 2012], sugere-se o acréscimo de valores de pressão, tamanho, tempo e velocidade para melhorar o método de autenticação já existente conhecido pelo nome de Password Patterns, que avalia apenas os gestos executados sob pontos pré-estabelecidos na tela. A proposta computou e comparou os resultados obtidos durante duas etapas em ambientes e períodos distintos, utilizando o sistema operacional Android. Na primeira etapa, os usuários precisaram executar 4 tipos pré-estabelecidos de gestos em momentos distintos para desbloquear a tela, em um período de dois dias. Enquanto na segunda etapa, os gestos eram mais complexos, com um tipo pré-definido de movimento, voltados para autenticação e durante um período de 21 dias. Com isso, os autores determinaram a influência do tempo sobre a capacidade do usuário em repetir movimentos, bem como, a melhora nos níveis de acuracidade do método após a inclusão de mais parâmetros de avaliação.

Em uma abordagem um tanto diferente, [GUERRA-CASANOVA et al. 2012], sugerem a empregabilidade do método de autenticação biométrico por gestos, através do uso de um dispositivo móvel como o iPhone 3G, onde as medições realizadas pelos sensores em três diferentes eixos são computadas e registradas a fim de formar uma espécie de assinatura no ar. Para isso, os autores criaram duas bases de dados de tamanhos e propósitos diferentes para que pudessem analisar desde a robustez do método diante de tentativas de falsificação até a dificuldade com a qual o usuário apresenta em repetir os gestos com o passar do tempo. Apesar dessa proposta não impor nenhum tipo de movimento pré-definido, os autores exigiram dos usuários a execução de três movimentos em diferentes direções para que uma amostra pudesse ser gerada. Por fim, passado um longo período de testes os autores puderam atestar a robustez do método que apresentou um percentual de 2,01% de Equal Error Rate (EER)[2]. Os testes também demonstraram uma pequena variação dos movimentos executados pelo usuário com o passar do tempo.

3. Experimento

O experimento proposto considerou os trabalhos realizados por [GUERRA-CASANOVA et al. 2012], [LUCA et al. 2012] e [BORAH 2012], ao definir a maneira pela qual seria conduzida esta etapa. Em [GUERRA-CASANOVA et al. 2012] é demonstrada a viabilidade e a forma como os movimentos executados por um indivíduo com um dispositivo móvel em mãos são medidos e registrados pelo acelerômetro do próprio aparelho. Em [LUCA et al. 2012] é sugerido o uso do algoritmo DTW para a geração de amostras, bem como a maneira pela qual mensurar taxas de falsa aceitação e falsa rejeição. Por fim, em [BORAH 2012] são apresentados os processos para a geração do modelo a ser utilizado durante a etapa de autenticação.

Para a realização do experimento foi necessário o desenvolvimento de um aplicativo na linguagem de programação Java para a plataforma Android. Entre as funcionalidades desenvolvidas estão: a configuração dos intervalos de tempo de captura do

[2]Segundo [MODI 2011], é o ponto onde as taxas de falsa aceitação e falsa rejeição são iguais

acelerômetro, a contabilização dos pontos capturados, o registro de amostras de acordo com a configuração estipulada, os cálculos de média e desvio padrão para a criação do modelo.

Para a execução do experimento foi definida uma amostra de 20 pessoas. A coleta dos dados foi realizada em dupla através de um único dispositivo móvel modelo Samsung Galaxy S4. Durante os testes, cada elemento da dupla desempenhou dois papéis: o usuário legítimo realizando a autenticação; e o impostor tentando burlar o processo de autenticação.

As etapas de registro e autenticação das quais o usuário legítimo realizou consistiram em:

- Selecionar uma das opções pré-estabelecidas de intervalo de tempo de captura do acelerômetro e uma das opções de movimento;
- Realizar o processo de registro repetindo o movimento desejado 5 vezes;
- Realizar três tentativas de autenticação

Ao final, o usuário impostor foi orientado a realizar três tentativas de autenticação executando o mesmo movimento e intervalo de captura entre pontos selecionados pelo usuário legítimo. Completado esses passos, o processo foi repetido posteriormente para os demais intervalos de tempo de captura serem concluídos. Posteriormente, outro movimento foi selecionado e os passos foram repetidos para cada intervalo de tempo de captura. Finalizado este processo, os papéis dos participantes foram invertidos.

A razão para divisão em duplas e as demais etapas de execução foram pensadas de modo a contabilizar as taxas de: falsa aceitação, quando uma tentativa de autenticação realizada pelo impostor é aceita como válida; e falsa rejeição, quando uma tentativa de autenticação do usuário legítimo é rejeitada. A decisão para um participante ser aceito ou não pelo sistema considerou o threshold, resultado da soma entre a média e o desvio padrão das comparações das amostras realizadas pelo algoritmo DTW.

A estrutura do experimento foi organizada nas seguintes etapas: configuração, captura, tratamento, registro e autenticação. Nas subseções a seguir são detalhadas cada uma delas.

3.1. Processo de Configuração

A primeira etapa do experimento requer o ajuste do intervalo de captura para uma das três opções: “tempo 1”, “tempo 2” e “tempo 3”, cujos valores foram definidos respectivamente como 20 milisegundos, 60 milisegundos e 200 milisegundos, baseado em [ANDROID 2013]. Além do intervalo de tempo de captura, o experimento considerou a realização de dois movimentos definidos neste trabalho como movimento simples e complexo. A definição de movimento simples foi baseada em um gesto com apenas uma curva. Já um movimento complexo foi associado a gestos com três ou mais curvas.

3.2. Processo de Captura

O processo para a obtenção dos dados necessários para as etapas de registro e autenticação tem início após a configuracão. A captura dos movimentos exercida pelo acelerômetro contido no dispositivo, compreende os deslocamentos realizados nos eixos: horizontal,vertical e aceleração, representados respectivamente pelas letras X, Y e Z em um

período de 4 segundos. Dessa forma, os intervalos de captura de 20 milisegundos, 60 milisegundos e 200 milisegundos, resultam respectivamente em um total de 200, 67 e 20 pontos coletados durante os 4 segundos.

Os resultados dessas execuções geram diferentes arquivos: amostra, contém os registros dos pontos capturados nos três eixos; e saída, relatório de pontos capturados nos eixos X,Y e Z de acordo com o registro de tempo.

3.3. Processo de Tratamento

Os dados obtidos durante a execução e armazenados em um arquivo de amostra identificam o comportamento do movimento sob um certo instante. Assim, para que o comportamento do movimento possa ser mensurado, faz-se necessário o uso de um algoritmo para analisá-lo. O experimento embasou-se nos trabalhos de [LUCA et al. 2012] e [BORAH 2012], para utilizar o algoritmo DTW para tratamento dos dados.

Conforme [LUCA et al. 2012] o comparativo entre duas amostras proporcionará um valor final, resultante da diferença da distância entre dois pontos de duas amostras. A definição sobre o número de amostras necessárias para a obtenção de um modelo a ser utilizado durante o processo de autenticação fundamentou-se em [BORAH 2012].

3.4. Processo de Registro

Obtidas as amostras, tem-se início o processo de formação do modelo. Com base no trabalho de [BORAH 2012], foram definidos como componentes formadores do modelo as duas amostras cujo resultado comparativo tenha sido o de menor valor.

Além disso, nesta etapa é calculado o threshold, resultado do desvio padrão dos comparativos entre as amostras acrescido do valor da média entre todas amostras. Esta forma de cálculo embasou-se em [LUCA et al. 2012].

3.5. Processo de Autenticação

Nesta fase, a amostra de autenticação é comparada com os dois arquivos indicados no modelo, isto é, com as duas amostras da fase de registro, cujo valor de comparação tenha sido o menor. Esses valores, por sua vez, são comparados ao threshold estabelecido durante a fase de registro para determinar se o usuário será ou não autenticado. Caso esses dois resultados apresentem valores inferiores ao threshold, o menor deles será computado.

4. Análise dos Resultados

Nesta seção serão apresentados e analisados os resultados provenientes do experimento, que computou 720 tentativas de autenticação de 20 usuários. Desse total de tentativas, metade foram de usuários legítimos e a outra de impostores, para que pudesse ser calculada as taxas de falsa rejeição e falsa aceitação, que respectivamente informam o percentual de indivíduos legítimos rejeitados e impostores aceitos pelo sistema. Nesta seção são utilizadas as siglas FRR (do inglês False Reject Rate) para apresentar as taxas de falsa rejeição e FAR (do inglês False Accept Rate) para apresentar as taxas de falsa aceitação. Visando uma melhor compreensão, a seção foi subdivida em: 4.1 Análise das Autenticações e 4.2 Taxas de Erros.

4.1. Análise das Autenticações

Para analisar as autenticações foi realizada a comparação do threshold de cada movimento com os valores obtidos durante as tentativas de autenticação, tanto do usuário legítimo como do impostor. Caso o valor obtido durante a fase de autenticação fosse inferior ao threshold, o usuário teria a sua tentativa de autenticação aceita.

Na Tabela 1 são demonstrados os resultados da comparação entre o valores das três tentativas de autenticação de um usuário legítimo considerando apenas o movimento simples e tempo 1. Como pode ser visualizado na Tabela 1, a primeira e a segunda tentativa de autenticação receberam o status aceito, pois os valores de autenticação (0,85 e 0,86) são inferiores aos valores do threshold (2,74). No entanto, a terceira tentativa retornou o status rejeitado devido ao valor de autenticação (3,25) ser superior ao threshold (2,74).

Tabela 1. Análise da aceitação ou rejeição da autenticação

Usuário	Movimento	Intervalo(ms)	Threshold	Autenticação	Status
Legítimo	Simples	20	2,74	0,85	aceito
Legítimo	Simples	20	2,74	0,86	aceito
Legítimo	Simples	20	2,74	3,25	rejeitado

Os resultados das tentativas de autenticação aceitas de todos os usuários legítimos podem ser visualizados na Tabela 2. O propósito ao expor estes dados em forma de tabela é realizar um comparativo através do qual as tentativas de autenticação aceitas pelos usuários possam ser comparadas nos três intervalos de tempo de captura e nos dois tipos de movimento.

Tabela 2. Tentativas de autenticação aceitas para o usuário legítimo

Movimento	Tempo 1 (20 ms)	Tempo 2 (60 ms)	Tempo 3 (200 ms)
Simples	58/60	52/60	50/60
Complexo	43/60	52/60	53/60

Observando a Tabela 2, percebe-se nos intervalos de tempo de captura do movimento simples ou mesmo onde a quantidade de pontos capturados é menor, um alto índice de aceitação. Em contrapartida, há uma maior dificuldade do usuário em ter a autenticação aceita quando executado um movimento complexo e um intervalo de tempo de captura com um número maior de pontos.

A análise das tentativas de personificação são apresentadas na Tabela 3. Confrontando os valores de rejeição dos intervalos de tempo de captura do movimento simples (33/60, 35/60 e 29/60) em relação aos apresentados no movimento complexo (47/60, 39/60 e 35/60), é possível verificar a dificuldade do impostor em personificar um usuário legítimo. Ainda, quando postos lado a lado os menores índices de rejeição (29/60 e 35/60) de cada um dos movimentos, percebe-se como uma menor captura de pontos torna o sistema mais vulnerável.

Tabela 3. Tentativas de autenticação rejeitadas para o usuário impostor

Movimento	Tempo 1 (20 ms)	Tempo 2 (60 ms)	Tempo 3 (200 ms)
Simples	33/60	35/60	29/60
Complexo	47/60	39/60	35/60

4.2. Taxas de Erros

A Tabela 4 apresenta as taxas de erros obtidas no experimento de modo a estabelecer um comparativo a fim de identificar quais intervalos de captura apresentam melhores níveis de segurança.

Tabela 4. Comparativo de taxas de falsa aceitação e falsa rejeição

Movimento	Intervalo(ms)	FRR(%)	FAR%
Simples	20	3,33	45,00
Simples	60	13,33	41,67
Simples	200	16,67	51,67
Complexo	20	28,33	21,67
Complexo	60	13,34	35,00
Complexo	200	11,67	41,67

Analisando as taxas de falsa aceitação apresentadas na Tabela 4 verifica-se que o uso de um movimento complexo e um intervalo de tempo de captura de 20 ms possui maior resistência em relação a tentativas fraudulentas de autenticação. Traçando um comparativo entre este resultado (21,67%) e os obtidos nos intervalos de tempo de captura de 60 ms e 200 ms do movimento complexo, onde as taxas de falsa aceitação foram respectivamente 35% e 41,67%, percebe-se um aumento de tentativas fraudulentas bem sucedidas. Isso é justificado em razão da quantidade de pontos capturados, pois uma vez que a quantidade de pontos diminui, as taxas de falsos positivos aumentam.

Essa mesma taxa de falsa aceitação do intervalo de tempo de captura de 20 ms do movimento complexo quando comparada com os 45% obtidos no mesmo intervalo de tempo de captura do movimento simples demonstra maior suscetibilidade a fraude quando realizado um movimento de menor complexidade. Em relação as taxas de falsa rejeição demonstradas, o melhor percentual foi visto também sob o intervalo de tempo de captura de 20 ms do movimento complexo. Este percentual demonstra que o uso de um número maior de pontos para autenticação também torna o sistema mais restrito ao usuário legítimo, pois este necessita repetir um movimento muito semelhante ao realizado durante o processo de registro.

5. Considerações Finais

O objetivo deste trabalho foi analisar a segurança de diferentes movimentos e intervalos de captura das coordenadas de um acelerômetro aplicados à biometria por gestos em dispositivos móveis. O experimento atestou a relevância do movimento em conjunto com o intervalo de tempo de captura em relação as taxas de falsa aceitação e falsa rejeição. Os valores de falsa aceitação e falsa rejeição notadamente sofreram variações quando a

complexidade do movimento, assim como, o intervalo de tempo de captura foram alterados. Como os dados demonstraram, para alcançarmos um nível melhor de segurança é necessário associar um grande número de captura de pontos com um movimento complexo. Esse aumento de segurança pode ser percebido através da alta taxa de FRR no movimento complexo e no intervalo de tempo de captura de 20 ms.

Os resultados obtidos também demonstraram que a redução do intervalo de tempo de captura de um acelerômetro e o aumento do grau de complexidade de um movimento causam a redução do número de tentativas de autenticações bem sucedidas para os usuários legítimos. Em contrapartida, sob um movimento de fácil execução e um intervalo de tempo de captura com menor capacidade em obter os pontos de um movimento, os níveis de aceitação foram superiores.

A maior dificuldade enfrentada neste projeto foi sem dúvida o período necessário para coleta de dados, bem como, a disponibilidade de participantes, uma vez que, cada um deles precisaria de 2 horas para completar todas as etapas, que iniciavam no registro, passavam pela autenticação e terminavam com tentativas de ataque ao sistema. Por essas razões, não foi realizada uma etapa de treinamento, para que o usuário pudesse se familiarizar com o sistema e provesse melhores resultados.

Como sugestão para trabalhos futuros indica-se a criação de um mecanismo para identificar amostras de baixa qualidade para mensurar falhas de registro e aquisição. Além disso, sugere-se a inserção de uma etapa de treinamento no processo de geração de uma amostra, de modo que o usuário possa familiarizar-se com os procedimentos a serem executados.

Referências

ANDROID, D. (2013). Sensors overview. http://developer.android.com/. Acessado em 2014-08-01.

BEN-ASHER, N., KIRSCHNICK, N., SIEGER, H., MEYER, J., BEN-OVED, A., and S., M. (2011). On the need for diferent security methods on mobile phones.

BONNEAU, J., PREIBUSCH, S., and ANDERSON, R. (2012). A Birthday Present Every Eleven Wallets? The Security of Customer-Chosen Banking PINs.

BORAH, P. (2012). Implementation of DTW Algorithm for Application Security.

GUERRA-CASANOVA, J., ÁVILA, C. S., BAILADOR, C., and SIERRA, A. D. S. (2012). Authentication in mobile devices through hand gesture recognition.

LUCA, A. D., HANG, A., BRUDY, F., LINDNER, C., and HUSMANN, H. (2012). Touch me once and I know it's you! Implicit Authentication base on Touch Screen Patterns.

MODI, S. K. (2011). *Biometrics in Indentity Management: Concepts to Applications*. Artech House.

SAE-BAE, N., AHMED, K., ISBISTER, K., and MEMON, N. (2012). Biometric-Rich Gestures: A novel Approach to Authentication on Multi-touch Devices.

SANOU, B. (2013). Ict facts and figures. http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf. Acessado em 2014-09-26.

Aplicações Seguras no uso de QR Code: Dois Estudos de Caso

Eduardo R. Costa[1], Jefferson O. Andrade[1], Karin S. Komati[1]

[1]Coordenação de Informática – Instituto Federal do Espírito Santo (Ifes) Campus Serra
Rodovia ES-010, Km 6.5 – 29.164-231 – Serra – ES – Brasil

erigamonte@gmail.com,{joandrade, kkomati}@ifes.edu.br

Abstract. *QR Codes simplify navigation in their mobile devices, however, this strategy exposes the users and even the system to several types of attacks, such as: fraud, cloned websites and SQL Injection. Therefore, security strategies should be adopted to avoid losses. On this context, the primary goal of this work is provide a QR Code safe utilization. This work presents two applications, the first is an association system between digitals and printed documents, which uses the message digest to verify the documents and keep the integrity of the association and, the second is a m-commerce system that uses encryption to maintain the secrecy of system input. Through the experiments results verified that these two strategies of security are effective when exists a malicious QR Code, making the graphic code utilization more safe and reliable.*

Resumo. *QR Codes facilitam a navegação de usuários em seus dispositivos móveis, entretanto, esta estratégia pode expor os usuários e o próprio sistema a diversos tipos de ataques, tais como: fraudes, sites clonados e injeção de SQL. Portanto, estratégias de segurança devem ser adotadas para evitar prejuízos. Neste contexto, o objetivo deste trabalho é apresentar propostas de utilização segura do QR Code. Este trabalho apresenta duas aplicações, a primeira é um sistema de associação entre documentos digitais e impressos que utiliza a verificação do resumo hash dos documentos para manter a integridade da associação e, a segunda um sistema de m-commerce que utiliza criptografia para manter o sigilo da entrada do sistema. Através dos resultados dos experimentos verificou-se que as duas estratégias de segurança propostas são eficazes quando existe um QR Code malicioso, tornando a utilização deste código gráfico mais segura e confiável.*

1. Introdução

Muitas vezes a navegação em um *smartphone* pode ser comprometida pelo tamanho da tela, além do fato de que muitas pessoas não possuem a destreza de digitar textos extensos nestes aparelhos. Para facilitar e agilizar o uso dos celulares, existe a tecnologia denominada de *QR Code* (***Q****uick* ***R****esponse Code* – Código de Resposta Rápida), que foi desenvolvido pela Denso Wave, lançado em 1994 e padronizado com o ISO/IEC 18004 [Denso Wave Incorporate, 2012].

O *QR Code* é uma imagem, um código de barras 2D, em duas cores, que pode ser lido e interpretado pelos *smartphones*. Quando a informação contida pelo *QR Code* for uma URL, direciona-se automaticamente para o endereço fornecido na internet, tornando o acesso fácil e rápido.

Uma característica visível do *QR Code* é a presença de três quadrados nos

vértices do código, chamados de padrões de localização, tal como pode ser visto na Figura 1. Esses padrões de detecção de posição garantem a leitura estável, evitando os efeitos negativos da interferência de fundo e possibilitando a leitura de 360º em alta velocidade. Enquanto os códigos de barras convencionais armazenam no máximo 107 dígitos, o *QR Code* pode armazenar até 7089 dígitos. As vantagens do QR Code são a grande capacidade de armazenamento, pouco espaço de impressão e a velocidade de leitura.

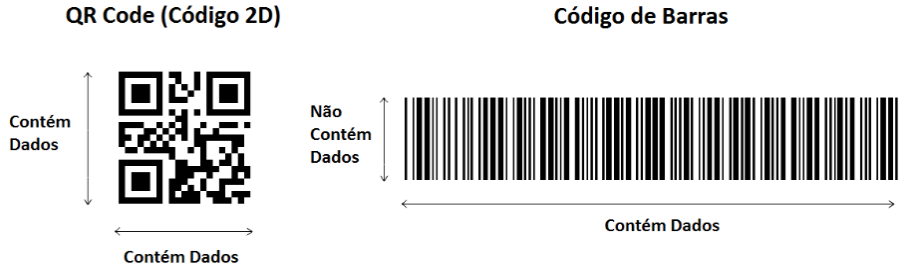


Figura 1. Exemplo de um QR Code e um código de barras. [Imagem Traduzida]. Fonte: [Denso Wave Incorporate, 2012]

O Banco do Brasil tem utilizado o *QR Code* para agilizar o pagamento de boletos [Banco do Brasil, 2011] pelo qual ganhou o troféu de inovação tecnológica do "Prêmio E-Finance 2011". Mais recentemente, este mesmo banco lançou o BB Code [Banco do Brasil, 2012], que é o uso do *QR Code* aliado à criptografia no qual o cliente autentica transações financeiras.

No entanto, a utilização desses códigos pode ser insegura caso não sejam tomadas algumas medidas. De acordo com Kieseberg e colegas (2010) existem várias formas de ataques utilizando *QR Code*, *u*m exemplo de uma ação mal intencionada seria o de substituir ou sobrepor o *QR Code* original, e com isso, redirecionar o usuário para outro site, que não o original. Um *QR Code* malicioso é um código normal, porém possui o conteúdo prejudicial.

Um caso real foi o que aconteceu na Rússia (Wasserman, 2011), no qual um *QR Code* adulterado enganou os consumidores que pensavam baixar um aplicativo Android chamado Jimm. O código baixado continha um *malware* que enviou códigos SMS para um número de telefone que cobrava por mensagem enviada, ocasionando prejuízos financeiros a muitas pessoas. Assim, medidas de proteção são necessárias para não haver vítimas.

Problemas envolvendo ataques através do QR Code são comuns e o motivo principal é que seres humanos não conseguem ler a informação que está codificada nele e por consequência não conseguem distinguir se o QR Code é malicioso ou não, antes de sua decodificação.

Nesse contexto, o objetivo deste trabalho é apresentar propostas de utilização segura do *QR Code*. O trabalho terá duas aplicações, a primeira é um sistema de associação entre documentos digitais e impressos (detalhada na seção 2) e a segunda é um sistema de compras rápido e seguro (detalhada na seção 3).

2. O Sistema Digital Document Stamp

A primeira aplicação, **Digital Document Stamp** é um sistema que faz uma associação entre a informação em papel e seu correspondente em formato digital, assegurando a

integridade desta associação. Esta proposta vem como uma solução para o armazenamento híbrido [Thomaz; Soares, 2004], isto é, a estratégia híbrida envolve a utilização de suportes convencionais (como o papel) e digitais, ao mesmo tempo.

Esta abordagem é utilizada quando o documento em papel for digitalizado e não pode ser destruído devido ao seu valor histórico ou legal. Caso a imposição seja pela legislação, o documento em papel só poderia ser destruído, desde que decorridos os prazos de guarda, decadência ou prescrição. É importante ressaltar que não é objetivo principal da digitalização a eliminação imediata do original, e sim, facilitar a disseminação e o acesso, além de evitar o manuseio do original, contribuindo para a sua preservação. Outra possibilidade de geração da abordagem híbrida é o documento ser criado na forma digital (nato-digital), porém a sua impressão em papel é necessária por questões de legislação ou arquivamento ou distribuição paga.

Uma solução possível para a estratégia híbrida é que a versão em papel possuísse um *QR Code* associado, onde bastaria que a pessoa usasse o seu celular (ou tablet) para ler este código e rapidamente seria realizado o *download* automático da versão digital.

A proposta é usar o *QR Code* para acessar rapidamente a informação digital associada, e também validar se o documento é exatamente o mesmo de quando foi feito o *upload* através do Hash Criptográfico. Hash Criptográfico é um código gerado a partir do conteúdo do documento. Caso um único bit seja alterado, adicionado ou retirado do conteúdo original, será gerado um código totalmente diferente [Tavares, 2006].

A estratégia híbrida traz um compromisso de que exista uma associação entre os dois documentos de mesmo conteúdo, mas em formatos distintos. Para verificar a integridade da associação, ou seja, garantir que aquilo que está para *download* é realmente o mesmo documento original que foi inicialmente disponibilizado no repositório remoto, basta ao terminar o *download*, calcular o código Hash Criptográfico e compará-lo com o código contido no *QR Code* e caso forem iguais, a associação entre a versão digital e a versão impressa é integra [Komati; Costa; Andrade, 2012], se não forem, o documento digital pode ter sido modificado ou o *download* pode tê-lo corrompido.

O sistema **Digital Document Stamp** possui dois subsistemas o gerador e o mobile, que são descritos a seguir.

2.1. Subsistema Gerador

Utilizaremos um *QR Code* para armazenar a URL e o código hash do documento. A Figura 2 ilustra como são os passos para a criação do documento digital e seu *QR Code* associado:

1. O documento em papel é escaneado;
2. Usuário informa em qual localização deve ser armazenado o documento digital;
3. Sistema gera o código hash deste arquivo;
4. Sistema gera um QR code com a localização e o código hash;
5. Imprime-se o QR Code, e guarda-se junto ao documento em papel (pode ser colado ao mesmo).

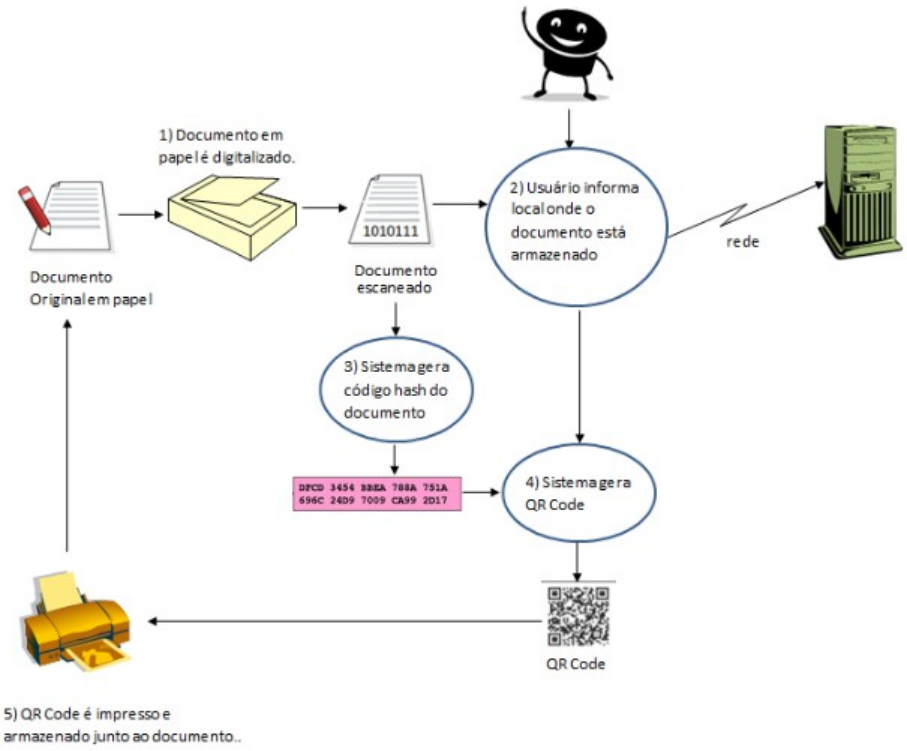


Figura 2: Digitalização de documento e criação de seu QR Code associado.

O subsistema Gerador é um sistema que é a junção dos passos 2, 3 e 4, isto é, todas as tarefas que se encontram descritas nas elipses da Figura 2. Para o desenvolvimento do subsistema Gerador usou-se Java 6 na IDE (Integrated Development Environment) Netbeans 7.1. Para a geração do código hash utilizou-se a classe MessageDigest que faz parte do pacote java.security. Esta classe MessageDigest gera códigos nos formatos: MD4, MD5, SHA-1 e SHA-512, que pode ser escolhido pelo usuário, para essa solução o formato escolhido foi o MD5.

2.2. Subsistema Mobile

Caso uma pessoa tenha acesso ao documento em papel que está com o *QR Code*, e queira, de forma rápida, o documento digital de mesmo conteúdo, esta pessoa usará o seu celular com o subsistema celular do sistema **Digital Document Stamp** instalado. A Figura 3 ilustra como são os passos para o fácil acesso e verificação de integridade do documento digital:

1. O usuário aponta o celular para o *QR Code*;
2. O sistema efetua a leitura do *QR Code*, decodificando as informações da localização do documento digital e do valor de hash;
3. O programa acessará a localização do documento digital e efetuar o download do arquivo;
4. Ao término do recebimento do documento, o programa calculará o código hash deste arquivo recebido;
5. O programa comparará se o código hash lido do *QR Code* é igual ao código calculado do arquivo recebido remotamente. Se os valores forem iguais, temos a garantia da integridade do documento.

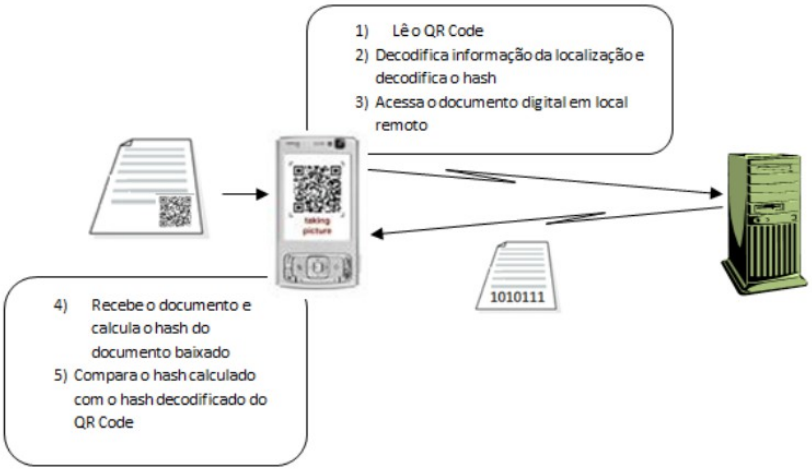


Figura 3: Acesso ao documento digital e verificação de integridade do mesmo.

No último passo, compara-se dois códigos hash: um código que foi decodificado da imagem do *QR Code* e outro calculado para o arquivo recebido. Caso sejam iguais, garante-se a integridade do documento. A integridade das informações é um aspecto que garante que os dados mantenham todas as características originais.

Caso os códigos sejam diferentes então ocorreu erro, onde há duas causas possíveis: ocorreu uma falha de transmissão ou o arquivo que se encontra remotamente não é o original. Caso a falha tenha sido durante a transmissão, provavelmente uma nova tentativa de acesso resolverá o problema. Se o documento digital que se encontra remotamente não foi o mesmo documento que gerou o *QR Code*, deve-se investigar se esta disparidade foi de origem maliciosa ou não. A investigação não faz parte da solução, esta solução só sinaliza a situação de não integridade.

É importante notar que o sistema de informação não controlará problemas de segurança de fator humano. Assim, se uma pessoa trocar o *QR Code* que está no documento em papel ou se o documento no servidor for alterado, o futuro usuário do subsistema Mobile não mais conseguirá acessar o documento digital correto.

O subsistema Mobile é um subsistema desenvolvido em Java 6 para o sistema Android, usando a IDE Eclipse versão Juno Mobile. Para a leitura do *QR Code* utilizou-se o ZXING [ZXing, 2013].

2.3. Experimentos e Resultados

Foram feitos três tipos diferentes de testes no sistema **Digital Document Stamp**:

- **Situação de integridade dos arquivos**: nesta situação, todo o processo é realizado e o arquivo baixado é exatamente o mesmo arquivo que gerou o *QR Code* testado.
- **Situação de associação não íntegra dos arquivos**: nesta situação, todo o processo é realizado e o arquivo baixado é diferente do arquivo que gerou o *QR Code* testado. Diferentemente do teste anterior, existe uma inconsistência na associação entre o documento que o *QR Code* referencia e o documento armazenado no servidor remoto. No momento da criação do *QR Code*, propositalmente, foi incluído um documento para calcular o código hash que não era o mesmo que o link indicava, gerando um código hash totalmente diferente.

- **Falha na rede durante o download**: durante o *download* provoca-se problemas na conexão de rede. Esta situação visa verificar o comportamento do sistema quando acontece uma faha na conexão de rede.

O sistema **Digital Document Stamp** obteve resultados satisfatórios. Todos os resultados esperados foram encontrados, tanto nas situações que possuíam a associação íntegra entre os documentos, quanto nos testes de falha: o teste que possuía a associação corrompida e no teste de falha na conexão. [1]

3. O Sistema Mobile Market

A segunda aplicação é um sistema de compras rápido utilizando a tecnologia *QR Code* e a ideia de vitrine virtual, o **Mobile Market** [Costa e Komati, 2013]. Uma vitrine virtual pode ser uma revista, folheto ou encarte, que em geral é distribuído gratuitamente ou pode estar na forma de cartazes dispostos em lugares públicos, como mostra a Figura 4.

O importante é que cada produto tenha um *QR Code* associado. A compra será entregue na residência do comprador, economizando o tempo de compra normal em um supermercado. O processo de compra pode ser feito, por exemplo, enquanto o usuário está no ônibus ou numa recepção, esperando ser atendido em uma consulta médica.

Figura 4: Acesso ao documento digital e verificação de integridade do mesmo.

Foram encontrados alguns sistemas similares ao **Mobile Market**, como o da Tesco [Estadão, 2012], da Seindor [Seindor, 2012] e do PicPay [PicPay, 2013]. Por motivos de segurança empresarial não foi encontrado, de forma detalhada, como estes sistemas implementaram as questões de segurança discutidas neste trabalho. Os códigos gráficos apresentados pela Tesco mostram um padrão com 4 estruturas nos cantos, ou seja, a Tesco desenvolveu seu próprio padrão de código gráfico. Nesta forma de segurança somente um grupo restrito conhece o processo de codificação e decodificação do código gráfico, no entanto, depende-se do sistema desenvolvido pela empresa. No sistema da empresa Seindor não foi encontrado nenhum padrão de segurança óbvio, mas foi encontrado uma vulnerabilidade. Os dados contidos nos *QR Codes* são todos links. Sem nenhuma dificuldade um atacante pode sobrepor o *QR Code* original do sistema e redirecionar o usuário para um site clonado e ter acesso a informações sigilosas. No sistema do PicPay, que não utiliza o *QR Code* como código gráfico e sim o Data Matrix, não foi possível encontrar um padrão óbvio de segurança, porém isso não quer dizer que não exista.

A proposta desta aplicação é apresentar uma solução segura contra ataques de

1 Confira vídeo de demonstração em https://db.tt/5QrGXRrB.

QR Codes maliciosos. A solução baseia-se na utilização *do QR C*ode com conteúdo criptografado. Inicialmente, o *QR Code* contendo o link e o preço do produto criptografados é criado, e posteriormente ele é associado a um produto na vitrine. Essa solução garante que, através do uso do aplicativo **Mobile Market**, o usuário está seguro de injeções de comandos, *phishing* e *pharming*, fraudes e outros tipos de ataques, além de garantir a autenticidade e a confiabilidade da informação.

O sistema é dividido em dois subsistemas: o Desktop, responsável pela cifragem dos dados e criação do *QR Code* (retângulo à esquerda da Figura 5) e o Mobile, que é o aplicativo responsável pela decodificação do código, decifragem da informação e realização do processo de compra (retângulo à direita da Figura 5).

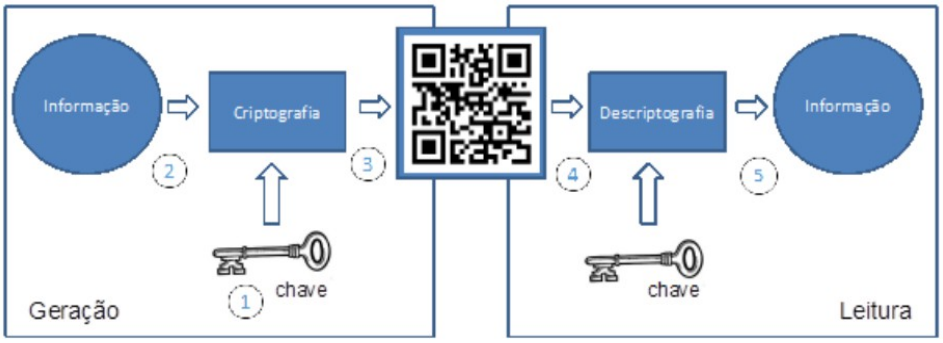


Figura 5: Fluxograma que mostra o passo a passo do processo de cifragem e decifragem da informação contida no QR Code.

O subsistema Desktop é responsável pela criação do *QR Code* para associar os produtos na vitrine virtual. O *QR Code* deverá conter no mínimo as seguintes informações: URL do produto no site para localizar o produto desejado no site da loja e o preço tal qual na vitrine. Uma das funcionalidades do módulo Desktop é a cifragem das informações e criação dos *QR Codes*, que serão associados na vitrine para a identificação no sistema do supermercado.

Esse módulo foi feito em Java usando a API QRGen [Gullaksen, 2012] para a geração do *QR Code*. Conforme a parte esquerda da Figura 5 denominada de "Geração", os dados do *QR Code* são cifrados utilizando, em um primeiro, a técnica de criptografia de chave simétrica com o algoritmo de *Advanced Encryption Standard* - AES 128 [CERT.BR, 2012].

O aplicativo Mobile faz a leitura do código, decodifica-o, e comunica-se com o site, conseguindo as informações sobre o produto, uma pequena imagem e o preço. De forma fácil, o usuário pode identificar quantos produtos ele deseja comprar ou mesmo se deseja retirar algum item do carrinho de compras. Ao finalizar o pedido, o usuário informará o endereço de entrega e uma forma de pagamento. Com o pagamento concluído, o aplicativo envia para o supermercado a lista de compra, os dados de entrega e os dados do pagamento.

Como é o aplicativo Mobile que fará a decodificação da informação gravada no *QR Code* e os dados no *QR Code* estarão cifrados, o usuário só conseguirá fazer as compras se usar o módulo Mobile fornecido, conforme a parte direita da Figura 5, denominada de "Leitura". Há o inconveniente de se fazer o *download* deste aplicativo no celular, mas será apenas uma vez.

O processo de criptografia é necessário para dar segurança ao usuário contra possíveis fraudes, pois caso não existisse a criptografia, o anúncio poderia ser alterado

com uma simples sobreposição do QR Code e, no ato da leitura, o usuário poderia ser redirecionado para um site clonado ou até um código malicioso ser instalado em seu *smarthphone*.

(a) Lista de Compras (b) Informações de Entrega

Figura 6: Imagens do sistema Mobile Market.

Na Subfigura 6a é apresentada a tela de montagem do carrinho de compras. Para adicionar um produto, o usuário deve clicar no sub-menu "Escanear Produto"e mirar o *smartphone* para o *QR Code* desejado, após a decodificação, o usuário poderá aumentar ou diminuir a quantidade de unidades que deseja comprar clicando nos símbolos de mais e menos localizados abaixo da foto do produto. Ao terminar de montar o carrinho, o usuário deve informar o endereço de entrega (Subfigura 6b) e depois clicar no sub-menu "Pagar com Paypal"para confirmar o pedido com o pagamento e enviá-lo para o supermercado. O módulo Mobile também foi desenvolvido na linguagem Java e utiliza a API Zxing para leitura de *QR Code*. Além disso, o módulo Mobile utiliza a API Mobile Payment [Paypal, 2012], que é uma biblioteca muito simples para que vendedores possam integrar pagamentos diretamente de suas aplicações.

3.1. Experimentos e Resultados

Os testes do sistema **Mobile Market** foram divididos em dois tipos: o primeiro de funcionamento e tem como objetivo verificar o fluxo normal do sistema, ou seja, testar a criação de uma lista de compras, e o segundo tem como objetivo mostrar a vulnerabilidade que o sistema enfrenta caso não possua sua informação protegida.

Este teste mostra que é possível a sobreposição do *QR Code*, ou seja, transformar um código que inicialmente é inofensivo em um código malicioso. Criou-se um *QR Code* com os dados que o sistema Mobile Market utiliza para seu funcionamento, porém os dados não estão cifrados, pois o objetivo é mostrar a vulnerabilidade que ocorre quando o atacante tem conhecimento de como é a sua entrada de dados do sistema. Criou-se um segundo *QR Code* que, além de possuir o texto anterior, também possui um código SQL Injection que tem o objetivo de trocar todos os preços dos produtos. O teste de sobreposição foi baseado em exemplos de MrReid [MRREID, 2012].

Neste teste foi utilizado uma versão do sistema **Mobile Market** que não tenta decifrar a informação, pois os *QR Codes* não foram criptografados. O resultado deste teste foi como o esperado, sem a utilização de uma proteção da informação contida no *QR Code* o sistema fica sujeito à ataques. O SQL *Injection* funcionou e modificou todos os preços dos produtos contidos no banco de dados do sistema.

4. Considerações Finais e Trabalhos Futuros

O uso de *smartphones* está cada vez maior e a sua evolução fez com que funcionalidades migrassem para a mobilidade de um celular, e uma das soluções para facilitar a interface humano-máquina é a utilização de um código gráfico e um dos mais populares é o *QR Code*. Problemas envolvendo ataques através d*o QR C*ode são comuns e o motivo principal é que seres humanos não conseguem ler a informação que está codificada nele e por consequência não conseguem distinguir se o *QR Code* é malicioso ou não, antes de sua decodificação. Assim, é importante se prevenir e manipular *QR Code* com a devida precaução, pois podem conter códigos maliciosos, fraudes ou redirecionamento indevido.

O principal objetivo desse trabalho foi estudar e propor formas seguras de utilização do código gráfico *QR Code*. Deve-se planejar e utilizar mecanismos de segurança para garantir que o uso de *QR Codes* não trará algum tipo de prejuízo para o usuário. Foram desenvolvidas duas aplicações diferentes com propostas de segurança diferentes: o **Digital Document Stamp** e o **Mobile Market.**

O primeiro propõe a utilização de hash criptográfico para a verificação de integridade da associação entre documentos digitais que são recuperados de um servidor remoto com o documento que foi inicialmente armazenado no local. Tem-se uma possível solução para validação da integridade e facilidade de acesso às informações, independente do formato do arquivo (texto, imagem ou vídeo). Todos os testes feitos tiveram resultados corretos, ou seja, foi possível verificar o comportamento do sistema quando a associação está integra ou corrompida ou quando não existe uma conexão com a internet. Assim, aproveitou-se o potencial de armazenamento do *QR Code* para armazenar o código Hash Criptográfico que fornecerá dados para a verificação da integridade da informação, validando se o documento digital que será acessado realmente corresponde ao *QR Code* associado ao documento em suporte convencional.

O segundo propõe a utilização de criptografia do conteúdo armazenado no *QR Code* para evitar que o utilizador seja vítima de *QR Codes* maliciosos. Desenvolveu-se um sistema de compras para dispositivos móveis para a validação da proposta. Todos os testes também obtiveram resultados corretos, onde foi possível montar uma lista de compras de uma forma segura, pois o sistema só utiliza a informação decodificada que ele consegue processar, qualquer outra informação é descartada. Um teste de sobreposição de *QR Code* também foi feito, no qual o código gráfico não estava cifrado e demonstrou-se o quanto o usuário fica vulnerável ao utilizá-lo.

Considera-se que a principal contribuição deste trabalho foi mostrar que tecnologias, que primeiramente podem parecer inofensivas, também devem utilizar formas de segurança. Este trabalho mostrou a utilização de segurança em dois sistemas específicos, porém as mesmas medidas podem ser utilizadas em várias outras aplicações, que possuem como entrada de dados a decodificação de códigos ou a transmissão de dados que estejam armazenados remotamente.

Como trabalhos futuros, no sistema **Mobile Market** pretende-se, utilizar outras estratégias de criptografia como a assimétrica e a híbrida para fazer um comparativo de resultados. No sistema **Digital Document Stamp** pode-se trocar o MD5 por outro método mais seguro SHA-256 ou mesmo deixar o próprio usuário selecionar o método. Além disso, pode-se acrescentar controle de acesso, solicitando login e senha, além de armazenar a identificação de quem fez o *upload* do documento no servidor. Realizar um estudo comparativo das vantagens, desvantagens e vulnerabilidades de ataques entre o

QR Code e outros tipos de códigos gráficos, como: Data Matrix, PDF 417, entre outros.

Referências

Banco do Brasil. (2011) “Clientes do BB já podem pagar boletos por leitura de QR Code”. http://www.bb.com.br/portalbb/page118,3366,3367,1,0,1,0.bb?codigoNoticia=31294

Banco do Brasil. (2012) “BB lança nova tecnologia de segurança para internet banking”. http://www.bb.com.br/portalbb/page118,3366,3367,1,0,1,0.bb?codigoMenu=&codigoNoticia=33936

CERT.BR (2012) “Cartiha de Segurança para Internet. 2012”. http://cartilha.cert.br/criptografia/

Costa, E. R.; Komati, K. S. (2013) “Um Sistema Móvel de Compras Rápido e Seguro via QR Code e Vitrine Virtual”. Em: X Encontro Anual de Computação (ENACOMP), Catalão, Goiás.

Denso Wave Incorporate. (2012) http://www.qrcode.com/en/index.html, Outubro.

Estadão. (2012). “A estratégia dos grandes: rede de supermercados inova ao colocar prateleira virtual no metrô. 2012”. http://pme.estadao.com.br/noticias/noticias,a-estrategia-dos-grandes-rede-de- supermercados-inova-ao-colocar-prateleira-virtual-no-metro,1708,0.htm

Gullaksen, K. (2012). "QRGen". http://kenglxn.github.com/QRGen

Kieseberg, P.; Leithner, M.; Mulazzani, M.; Munroe, L.; Schrittwieser, S.; Sinha, M. e Weippl, E. (2010) “Qr Code Security”. In: Proc. of The International Workshop on Trustworthy Ubiquitous Computing (TwUC’10), Paris, France.

Komati, K. S.; Costa, E. R.; Andrade, J. O. (2012) “Gerenciamento de Informações com QR Code e Código Hash Criptográfico”. Em: XIX Simpósio de Engenharia de Produção (SIMPEP), Bauru, São Paulo, Brasil.

MRREID (2012). "Hacking QR codes". http://wordpress.mrreid.org/2011/08/06/hacking-qr-codes/

Paypal (2012). "Mobile Payment Libraries". https://www.paypal-brasil.com.br/x/xblog/2012/02/15/mobile-payment- libraries

PicPay (2013). “Pagar com PicPay é muito melhor”https://www.picpay.com/

Seindor (2012) “Virtual Store”. https://seindor.com/virtualstore/

Tavares, P. H. N. (2006) “Estudo e implementacao de algoritmos de resumo (hash) criptográfico na plataforma Intel XScale”. Monografia (Mestrado). UNICAMP.

Thomaz, K. P.; Soares, A. J. (2004) A preservação digital e o modelo de referência Open Archival Information System (OAIS). Em: *DataGrama Zero - Revista de Ciência da Informação*, v. 5, n. 1.

Wasserman, T. (2011) “New Security Threat: Infected QR Codes”. http://mashable.com/2011/10/20/qr-code-security- threat/.

ZXing (2013). ZXing (’Zebra Crossing’). http://code.google.com/p/zxing

Implementação em Hardware de Instrução Segura de Acesso à Memória - Caso MIPS 16 bit

Eric S. Torres[1], **Antonio L. Maia Neto**[1], **Omar P. Vilela Neto**[1], **Leonardo B. Oliveira**[1]

[1] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG, Brasil

{eric.torres,lemosmaia,omar,leob}@dcc.ufmg.br

Abstract. *Some languages do not have any security mechanisms. This causes many programs exposed to attacks. A common attack technique is Buffer Overflow. There are several proposed solutions to this problem. However, most of these solutions are implemented in software, causing overhead. Therefore, this paper proposes a hardware alternative, able to perform safe memory access. For this purpose, it was created the SSW, a safe memory access instruction for MIPS 16 bit.*

Resumo. *Algumas linguagens não possuem mecanismos de segurança. Isso faz com que diversos programas estejam vulneráveis a ataques. Um ataque bastante comum é o Buffer Overflow. Existem diversas propostas de solução para esse problema. Mas, em sua grande maioria, essas soluções são implementadas em software, gerando uma grande sobrecarga. Por isto, este trabalho propõe uma alternativa em hardware, capaz de realizar o acesso seguro à memória. Foi criado então a SSW, uma instrução segura de escrita à memória desenvolvida para a arquitetura MIPS 16 bit.*

1. Introdução

No contexto atual, uma das maiores preocupações é a Segurança de Software - SS. Uma aplicação segura deve garantir que seu funcionamento será sempre conforme esperado e que as informações que o mesmo transita devem ter sua integridade e sua confidencialidade mantidas.

Porém algumas linguagens de programação, como por exemplo, a linguagem C, não possuem nenhum mecanismo de segurança. Isso se deve, na maioria das vezes, à concepção da linguagem com o foco na eficiência do uso de recursos, isto é, a linguagem deve interferir o mínimo possível em questões de execução. Isto acarreta em que o programador deve inserir explicitamente os mecanismos de segurança. Do contrario o software produzido poderia conter diversas vulnerabilidades.

As vulnerabilidades de um sistema de software podem levar a diversos tipos de ataques. Um dos ataques mais devastadores é o ataque de *Buffer Overflow* - BOF. Esse ataque é realizado com sucesso quando, através da escrita de dados em um arranjo, a quantidade de dados inseridos é maior que a capacidade do arranjo. Consequentemente, posições de memória adjacentes ao arranjo serão sobrescritas com dados inseridos pelo adversário. Essa situação pode acarretar no funcionamento incorreto do sistema ou, em alguns casos, alteração do fluxo de execução do programa. O *worm* Morris[1], por exemplo,

[1] http://en.wikipedia.org/wiki/Morris_worm

explorava uma vulnerabilidade de BOF para executar um ataque de DoS. Apesar desse ataque ser bastante conhecido hoje em dia, ainda é uma grande ameaça [Cowan et al. 1998].

Para evitar este tipo de ataque, é preciso realizar a Verificação de Limites de Arranjo (*Array-Bounds Check* – ABC). Isto pode ser feito manualmente pelo programador, ou inserido através de ferramentas automatizadas. Apesar de eficaz, essa solução se mostra ineficiente, devido à grande sobrecarga acarretada pela solução em software.

Este trabalho vem apresentar a construção de uma forma de acesso seguro à memória através de uma instrução de hardware. Esta instrução garante que o todo endereço de memória acessado esteja dentro do limite superior do arranjo alvo.

2. Embasamento Teórico

Nesta seção serão apresentados alguns conceitos importantes para o desenvolvimento deste trabalho. Na seção 2.1 são apresentados alguns conceitos de SS, introduzindo técnicas de ataque e aprofundando um pouco mais sobre o ataque BOF. Já seção 2.2 se dedica à análise dos ABCs e seu impacto nos programas. Finalmente, a seção 2.3 faz uma pequena introdução ao MIPS, a arquitetura utilizada para o desenvolvimento do trabalho.

2.1. Segurança de Software

No contexto de SS, vulnerabilidade é uma falha em um software que pode ser utilizada por um adversário para executar ações maliciosas. Estas ações podem comprometer o funcionamento do software e da máquina que o executa, expor informações confidenciais, ou ainda, corromper dados sigilosos.

Como os ataques que exploram estas vulnerabilidades tem crescido de maneira vertiginosa, SS se torna um ponto crucial no desenvolvimento de sistemas computacionais.

Os ataques poder ser divididos em quatro categorias sendo elas:

- Sigilo: revelam informações confidenciais do programa ou do próprio ambiente de execução, como, por exemplo, endereço de arranjos.
- Integridade: manipulam os dados para serem utilizados em um contexto diferente do que originalmente deveriam pertencer. Dentre os ataques mais comuns estão o Integer Overflow e o BOF.
- Disponibilidade: tornam algum tipo de serviço indisponível para o uso.
- Autenticidade: acessam/disponibilizam informação da qual o atacante não possui o privilégio de acesso.

Este trabalho tem como foco principal os ataques de integridade, mais especificamente o ataque BOF, que será detalhado a seguir.

2.1.1. Buffer Overflow

O ataque de BOF é um ataque de integridade, que faz uso indevido do arranjo. Entendendo arranjos como dados contíguos da memória, o BOF visa sobrescrever os dados adjacentes, que não pertençam às posições de memória do arranjo alvo, até que atinjam

seu objetivo. As consequências deste ataque variam de funcionamento incorreto do software a comprometimento do ambiente de execução.

Para entender os riscos desse ataque, é preciso explicar a divisão de memória de um processo. A divisão é mostrada abaixo e ilustrada na Figura 1:

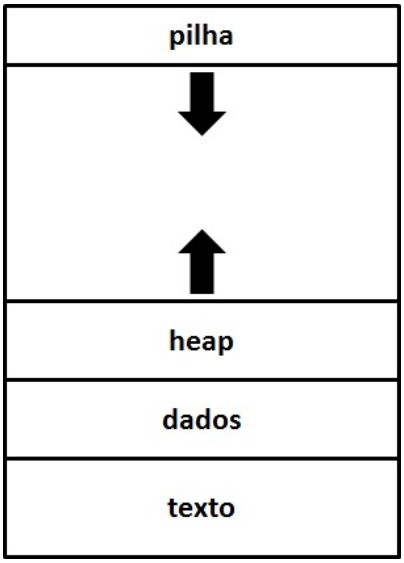


Figura 1. Divisão da memória de um processo.

- **Texto** - região de memória estática, de tamanho fixo, que contém o código do programa.
- **Dados** - região de memória estática, onde são armazenadas as constantes e variáveis globais.
- ***Heap*** - região dinâmica de memória, usada para armazenar as porções de memória alocadas dinamicamente pelo programa.
- **Pilha** - região dinâmica e contígua de memória, utilizada para armazenar variáveis de controle e realizar a troca de contexto entre procedimentos.

Os ataques BOF podem acontecer de duas maneiras:

- Ataques baseados na pilha - neste tipo de ataque podem ser sobrescritas variáveis de controle como o endereço de retorno ou o *stack pointer*. Em ambos os casos, o fluxo de execução é alterado de acordo com a vontade do adversário, podendo expor o sistema a códigos maliciosos.
- Ataques baseados em *heap* - ocorrem geralmente através da corrupção das estruturas de controle, para assim sobrescrever o dado desejado, como o endereço de retorno [Robertson et al. 2003].

Uma das maneiras de se evitar esse ataque é através de ABCs, que serão analisados na seção seguinte.

2.2. Array-Bouds Check

Uma forma de defesa contra ataques de BOF são os ABCs. Esta técnica consiste em averiguar os limites do arranjo antes de se fazer o acesso à memória. A Figura 2 mostra um código vulnerável (esquerda) e como o ABC é construído para o caso (direita).


```
...
int buffer[MAX];
int i,j,a;
...
for(i = 0; i < j; i++)
{
        ...
        buffer[i] = a;
        ...
}
```

```
...
int buffer[MAX];
int i,j,a;
...
for(i = 0; i < j; i++)
{
        ...
        if(i > 0 && i < MAX)
                buffer[i] = a;
        ...
}
```

Figura 2. Código Vulnerável x Código utilizando ABC.

Esta técnica, porém, gera uma sobrecarga de código. A Figura 3 mostra um paralelo entre o código *assembly* para a arquitetura MIPS do programa vulnerável (esquerda) e o programa com o ABC em software. Como observado são necessárias pelo menos duas instruções de *branch* condicional a mais para cada acesso à memória, o que pode acarretar uma grande perda de desempenho. É preciso também calcular os endereços limites do arranjo para fazer a comparação.

```
LOOP:
        ...
        sw $R1,0($R2)
        ..
        jmp LOOP
```

```
        addi $R3, $R0, MAX;   tamanho do array
        add  $R4, $R2, $R3;   $R2 contém o endereço base do arranjo
LOOP:
        ...
        bgt  $R2, $R4, AFTER
        bgt  $R0, $R2, AFTER
        sw $R1,0($R2)
AFTER:
        ...
        jmp LOOP
```

Figura 3. Código MIPS vulnerável(esquerda) e implementando Array-Bound Check (direita).

2.3. MIPS

MIPS é um microprocessador desenvolvido pela *Stanford University* entre 1982 e 1984. Ele é baseado na arquitetura RISC [Gross et al. 1988] e apresenta características como: tamanho fixo de instrução e *pipeline* em nível de hardware. Mais informações sobre sobre o projeto da arquitetura estão públicas [Patterson and Hennessy 2008].

2.3.1. MIPS 16 bit

A arquitetura MIPS 16 *bit* é uma simplificação do MIPS desenvolvido pela *Stanford University*. Isso implica em um número reduzido de instruções, menor número de registradores e capacidade de endereçamento limitada.

Nessa arquitetura o espaço de memória principal está limitada a capacidade de endereçamento do processador, por isto, são utilizados apenas 64KB de dados. Além

disto o número de registradores também é limitado a 16, devido às características das instruções da arquitetura.

As instruções são construídas a partir de uma palavra, com 16 *bits* para representá-las. Destes 16 *bits*, 4 são utilizados para representar o *opcode*, que indica qual a instrução em questão. As instruções podem ser divididas em três tipos:

- Tipo-R - instruções que necessitam três registradores. Cada registrador é identificado por um conjunto de 4 *bits*. Grande parte das instruções aritméticas é deste tipo.
- Tipo-I - instruções que necessitam de dois registradores e uma constante, sendo utilizados 4 *bits* paracada registrador e mais 4 *bits* para a constante. As instruções de acesso à memória são deste tipo.
- Tipo-J - instruções que não necessitam registradores. Nesse caso todos os 12 *bits* são utilizados para representar constantes.

As instruções implementadas na arquitetura escolhida para o desenvolvimento deste trabalho são: `add`, `sub`, `addi`, `and`, `or`, `not`, `shiftl`, `shiftr`, `lw`, `sw`, `beq`, `jump`, `halt` e `nop`.

Além disto, esta arquitetura utiliza 5 estágios:

- *Fetch* - nessa etapa a instrução corrente, determinada pelo Contador de Programa (*Program Counter* – PC), é lida da memória e armazenada no Registrador de Instrução (*Instruction Register* – IR). Em seguida, o PC é atualizado adequadamente.
- *Decode* - etapa de decodificação da instrução previamente lida. É nesse estágio que são definidos e armazenados todos os sinais de controle. Além disso, são armazenadas as informações sobre os registradores ou dados que a instrução deverá considerar nos estágios seguintes.
- *Execute* - nesta fase a Unidade Lógica Aritmética (*Aritmetic Logic Unit* – ALU) faz a computação requerida pela instrução.
- *Memory* - o acesso à memória, para a leitura ou escrita, é realizado nessa etapa.
- *Write-back* - nesse estágio é concluída a execução da instrução. No caso, o dado lido da memória ou o resultado da operação executada pela ALU é escrito no registrador indicado pela instrução.

3. Trabalhos Relacionados

A proteção de sistemas computacionais contra ataques de BOF é abordada por diversos trabalhos ([Wilander and Kamkar 2003,Serebryany et al. 2012]). As propostas de defesas subdividem-se, na maior parte dos casos, em Análise Estática e Análise Dinâmica de código.

A Análise Estática de código [Viega et al. 2000] é aplicada durante o desenvolvimento do software. Nessa etapa o objetivo é identificar possíveis trechos vulneráveis a partir de diferentes técnicas.

Já a Análise Dinâmica [Cowan et al. 1998] é caracterizada pela aplicação de diferentes técnicas de instrumentação de código. Em alguns casos, o código instrumentado pode lançar mão de funcionalidades presentes no hardware do dispositivo. No trabalho [Shao et al. 2005], por exemplo, é apresentada a implementação de uma instrução

dedicada à verificação em hardware de limites de arranjos para a arquitetura DLX. A instrução proposta é formada por três operandos registradores: um registrador que deve manter o endereço de acesso ao arranjo e dois outros registradores que guardam seus limites inferior e superior. Quando a instrução é executada, o endereço de acesso é comparado com os limites e, caso seja detectado uma tentativa de violação de memória, um sinal de interrupção é lançado. Caso contrário a próxima instrução, de acesso à memória, pode ser executada.

A nossa solução se difere da proposta citada na tentativa de tornar os programas mais eficientes. Nesse sentido, abrimos mão da verificação do limite inferior dos arranjos para que, em uma única instrução, haja a possibilidade de verificar o limite superior e concluir a escrita na memória. Essa estratégia foi adotada ao notar que os ataques bem sucedidos a programas com vulnerabilidades de BOF são mais comuns devido a falta de verificação dos limites superiores.

4. Safe Store Word

Nesta seção serão descritas algumas decisões tomadas durante a execução desse trabalho. Apresentaremos, também, o projeto da instrução segura e como foi realizado o desenvolvimento da mesma.

4.1. Decisões de projeto

Este trabalho tem como objetivo principal provar a eficácia da realização do ABC via hardware. Por isto foi decido construir apenas a instrução de escrita na memória, já que apenas esta seria suficiente para inibir o ataque BOF. A instrução de leitura segue os mesmos princípios aqui apresentados, o que possibilita sua futura concepção.

A outra decisão, brevemente discutida na seção 3, é quanto a verificação de apenas um dos limites do arranjo durante uma operação de escrita na memória. Para isso, é importante notar que um ABC em hardware seguido de uma escrita na memória demanda quatro informações: endereço inicial do arranjo, endereço do limite superior do arranjo, endereço de escrita e o dado a ser gravado na memória. Contudo, a arquitetura MIPS, tomada como base para o projeto, tem instruções de tamanho fixo, o que impede o uso de mais de três registradores. Assim, foi decidido queo limite a ser verificado é o superior, já que é onde as vulnerabilidades de BOF são mais exploradas.

Outro ponto a ser discutido é o lançamento de interrupção. Para que os programas mantenham um estado consistente após um acesso a memória é necessário que o processador indique que um acesso não pôde ser concluído. Por isso, durante a implementação real da instução de escrita segura deve ser considerada uma interrupção que indique acesso fora dos limites.

Feitas essas considerações, foi concebida a instrução de escrita segura (*Safe Store Word – SSW*). Seu funcionamento é discutido a seguir.

4.2. Projeto

A instrução SSW é a instrução de escrita segura na memória. Diferente das instruções MIPS tradicionais de acesso à memória, SSW é uma instrução do Tipo-R, já que são necessários três registradores. Os detalhes do projeto da instrução são apresentados na Figura 4 e detalhados a seguir:

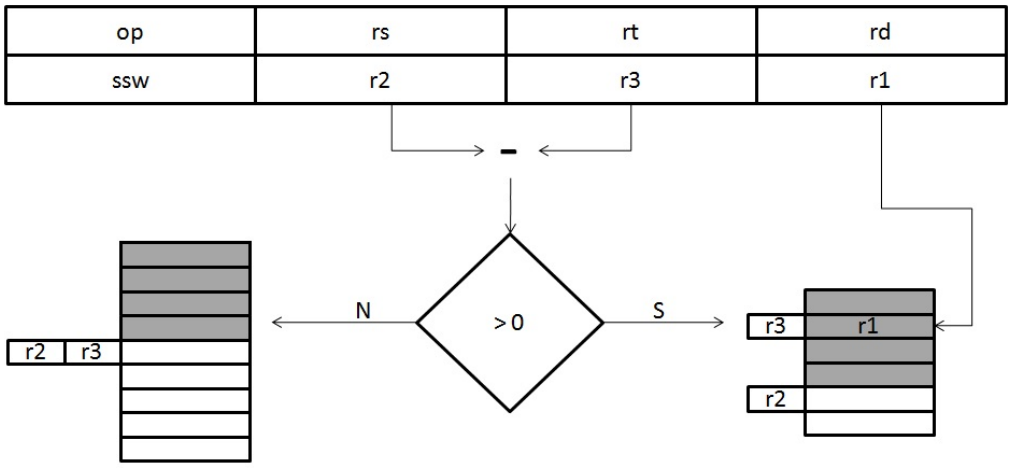


Figura 4. Forma de utilização da instrução safe store word.

- Registradores - para esta instrução são necessários 3 registradores. O primeiro registrador indica o limite superior do arranjo, o segundo contém o endereço de escrita e o terceiro mantém o dado que deve ser escrito na memória.
- Comparação - a comparação feita para validar a escrita na memória segue um princípio bem simples. É realizada a subtração entre o endereço do limite superior e o endereço a ser acessado da memória. Caso o resultado seja maior que zero o endereço de acesso respeita o limite superior, caso contrário a escrita do dado não deve ocorrer.
- Tratamento - em caso de falha na verificação, será bloqueada a escrita na memória e o processador levanta uma interrupção para indicar o erro.

Essa instrução utiliza 4 estágios de pipeline, sendo os estágios de execução mostrados na Tabela 1.

Tabela 1. Estágios de Pipeline.

Estágio do pipeline	Operações
Fetch	$IR = MEM[PC]$ $PC = PC + 1$
Decode	$A = rs$ $B = rt$ $C = rd$
Execute	$A - B$
Memory	$if((A - B) > 0)$ MEM[B] $= C$ $else$ INT
Write-back	-

4.3. Desenvolvimento

O desenvolvimento foi realizado em duas fases. A primeira consiste na codificação do caminho de dados do MIPS 16 *bit* em Linguagem de Descrição de Hardware (*Hardware Description Language* – HDL), mantendo todas as características da arquitetura.

Esta fase foi concluída produzindo um código de 1663 linhas em Verilog. Este material foi sintetizado fisicamente na placa Cyclone II da Altera. A Figura 5 mostra o *datapath* gerado a partir da implementação realizada.

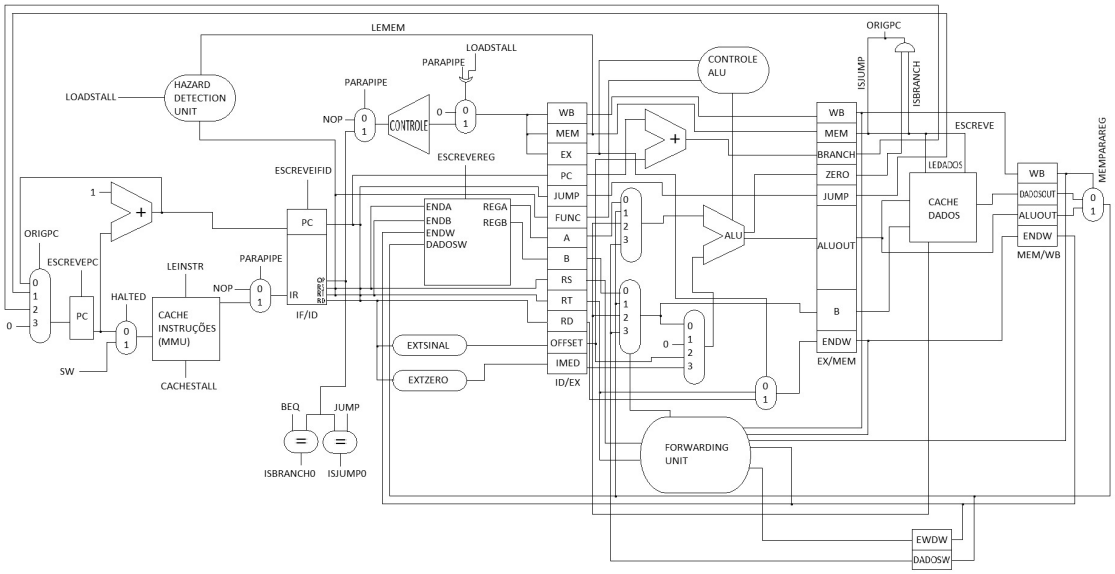


Figura 5. *DataPath* original.

A segunda fase englobou a alteração dos módulos necessários para a construção da nova instrução, adicionando fios, registradores e novos módulos.

Após concluída esta fase, o código apresentava 1710 linhas de código, um acréscimo de apenas 2.83% de linhas de código. A sobrecarga de elementos lógicos reportada pela ferramenta de síntese foi de apenas 2.43%. A Figura 6 mostra o *datapath* resultante das modificações. Em vermelho são destacadas as modificações necessárias para o desenvolvimento da nova instrução.

5. Análise

Esta seção descreve a forma de avaliação da nova instrução, apresentamos a metodologia da avaliação na seção 5.1 e os resultados obtidos na seção 5.2.

5.1. Metodologia

A análise da eficácia e do desempenho da instrução segura foi feito através da comparação de três versões de um mesmo programa:

- Original - os arranjos são acessados sem nenhum ABC.
- *Baseline* - é utilizado um ABC em software (verificando apenas o limite superior).
- SSW - o programa será modificado de forma que utilize a nova instrução.

Os programas foram executados na placa *Cyclone II* e o número de instruções necessárias para a execução foi obtido.

A seguir são descritos os algoritmos que foram usados como teste.

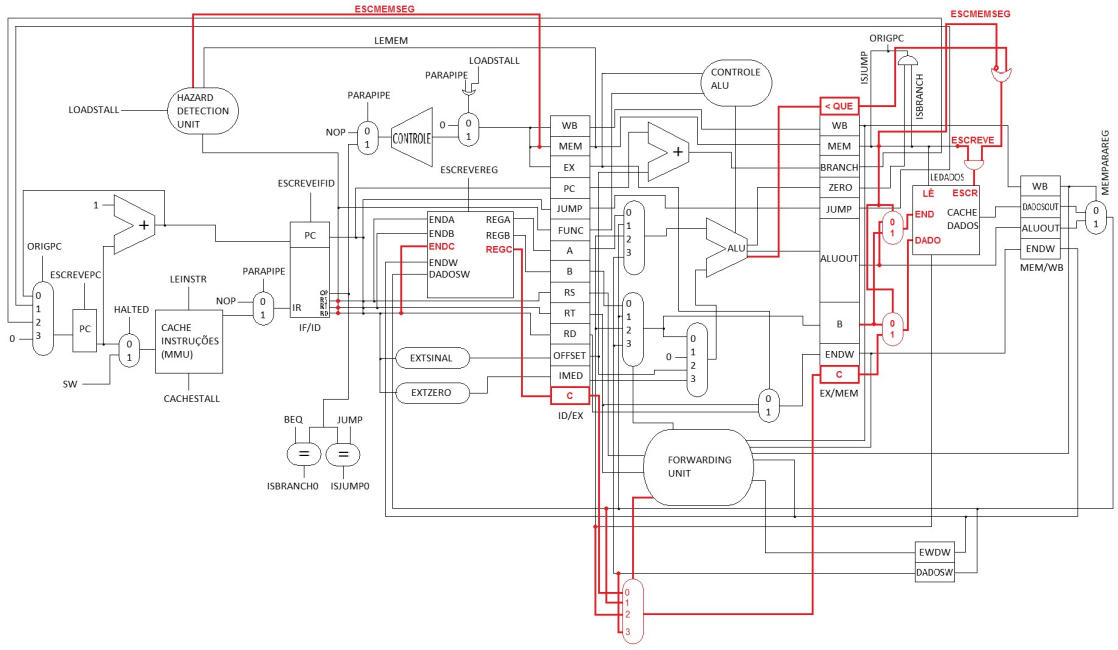


Figura 6. *DataPath* após a modificação.

5.1.1. Algoritmo

Para a validação deste trabalho foi desenvolvido um teste de *copyarray*. Este teste consiste na construção de dois arrays, um de tamanho 32 *bytes* e o outro de tamanho 16 *bytes*. O array de 32 *bytes* é iniciado com a palavra CODE em todas as suas posições. É realizado então a cópia *byte a byte* do arranjo maior para o menor. Durante a fase de testes também não foi considerada o lançamento de interrupção no caso de tentativa de acesso fora dos limites, deixamos que a instrução segura simplesmente não concluísse a escrita.

Com este método espera-se que o código original não proteja o espaço de memória adjacente ao segundo arranjo. Por outro lado, as versões *baseline* e SSW devem escrever apenas na região válida.

5.2. Resultados

Nesta seção serão apresentados e discutidos os resultados alcançados.

Na Tabela 2 a primeira coluna identifica o programa utilizado. A coluna seguinte apresenta o número total de instruções necessárias para a execução do programa. Por fim, é mostrado a porcentagem de sobrecarga no número de instruções executadas em relação ao programa original. Como pode ser observado, o *baseline* apresenta uma sobrecarga muito alta, o que não ocorre na solução utilizando SSW. Isso se deve ao fato de que a única informação necessária para o funcionamento da nova instrução é o calculo do limite superior do arranjo. Já a solução de ABC em software precisa realizar desvio condicional.

6. Trabalhos Futuros

Como trabalhos futuros é preciso construir uma forma de validação do limite inferior e, também, conceber a instrução de leitura segura (*Safe Load Word* – SLW). Posteriormente

arraycopy	**Instruções**	**Sobrecarga(%)**
original	282	
baseline	346	22,7
SSW	288	2,1

Tabela 2. Resultados dos testes.

deve ser avaliado o desempenho das novas instruções adotando metodologia semelhante à aqui apresentada.

Referências

Cowan, C., Pu, C., Maier, D., Hinton, H., Walpole, J., Bakke, P., Beattie, S., Grier, A., Wagle, P., Zhang, Q., et al. (1998). Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks. In *the 7th USENIX Security Symposium*, pages 346–355.

Gross, T. R., Hennessy, J. L., Przybylski, S. A., and Rowen, C. (1988). Measurement and evaluation of the mips architecture and processor. *ACM Trans. Comput. Syst.*, 6(3):229–257.

Patterson, D. A. and Hennessy, J. L. (2008). *Computer organization and design: the hardware/software interface*. Morgan Kaufmann.

Robertson, W. K., Kruegel, C., Mutz, D., and Valeur, F. (2003). Run-time detection of heap-based overflows. In *LISA*, volume 3, pages 51–60.

Serebryany, K., Bruening, D., Potapenko, A., and Vyukov, D. (2012). Addresssanitizer: a fast address sanity checker. In *Usenix Annual Technical Conference (ATC'12)*, pages 28–28.

Shao, Z., Xue, C., Zhuge, Q., Sha, E. H.-M., and Xiao, B. (2005). Efficient array & pointer bound checking against buffer overflow attacks via hardware/software. In *International Conference on Information Technology: Coding and Computing, 2005 ITCC'05.*, volume 1, pages 780–785. IEEE.

Viega, J., Bloch, J.-T., Kohno, Y., and McGraw, G. (2000). ITS4: A static vulnerability scanner for c and c++ code. In *the 16th Computer Security Applications (ACSAC'00)*, pages 257–267.

Wilander, J. and Kamkar, M. (2003). A comparison of publicly available tools for dynamic buffer overflow prevention. In *the 10th Annual Network and Distributed System Security Symposium (NDSS'03)*.

Esteno[*]: Uma Abordagem para Detecção Visual de *Bankers*

Victor F. Martins[1], André R. A. Grégio[1,2], Vitor M. Afonso[1], Paulo Lício de Geus[1]

[1]Universidade Estadual de Campinas (Unicamp) – Campinas – SP – Brasil

[2]Centro de Tecnologia da Informação Renato Archer (CTI)– Campinas – SP – Brasil

{furuse, vitor, paulo}@las.ic.unicamp.br, andre.gregio@cti.gov.br

Abstract. *Bankers—Internet Banking information stealer programs—usually present windows that mimic legitimate bank sites to lure users into providing sensitive information. In addition, bankers may run on the target operating system in a non-intrusive mode, making the detection and analysis provided by unsupervised, automated analysis systems difficult. In this paper, we propose a solution for the identification of Brazilian bankers. To this end, we leverage three visual analyzers (based on color properties, known logotype presence and textual patterns) that are tuned using a supervised machine learning technique (Random Forest). We tested our approach on over 1,100 unknown binaries´ images, yielding 92.1% of correctly classified samples.*

Resumo. *Bankers—programas maliciosos para roubo de informações bancárias—geralmente usam janelas que imitam sites dos bancos reais para ludibriar os usuários. Eles podem atuar de maneira não intrusiva no sistema alvo, o que dificulta a detecção e análise por sistemas automáticos não supervisionados. Neste artigo, apresenta-se uma proposta de solução para a identificação de bankers brasileiros. Para tanto, aplica-se três analisadores visuais (cores, presença de logotipos de banco e conteúdo de textos) refinados usando aprendizado de máquina supervisionado (Random Forest). Testes com mais de 1.100 imagens extraídas de binários desconhecidos resultaram em 92,1% de exemplares corretamente classificados.*

1. Introdução

A sociedade atual tem migrado para o espaço virtual, do comércio aos relacionamentos interpessoais. Consequentemente, tem sido crescente a utilização de *Internet Banking* para a realização de transações financeiras, como pagamentos e transferências. Com isso, aumentou também a motivação dos atacantes para roubar credenciais bancárias utilizadas pelos clientes no acesso via Internet. Logo, surgiram programas maliciosos cujo principal objetivo é ludibriar os usuários visando obter suas credenciais. Este tipo de *malware* ficou conhecido como *crimeware*, *information stealer*, *phishing Trojan*, *banking Trojan* ou simplesmente *banker* [Corporation 2007], no caso mais específico que é o foco deste artigo.

Bankers geralmente aplicam técnicas de engenharia social para levar um usuário a fornecer os dados de acesso a sua conta bancária na Internet, tais como a agência, conta corrente, nome de *login* e senha, número do cartão de crédito ou débito e valores de tabelas de senhas ou *tokens* de segurança. As informações coletadas são então enviadas ao atacante, podendo ser vendidas ou utilizadas para o pagamento de contas e compras não autorizadas ou, ainda, podendo ocorrer a transferência de dinheiro da vítima para contas de terceiros. Existem *bankers* em todos os países que adotam

[*]*Esteno* é uma Górgona da mitologia grega, irmã de Medusa.

plataformas de *Internet Banking*, porém, no Brasil, os ataques e prejuízos atingem cifras alarmantes, devido ao avanço deste tipo de tecnologia no país em decorrência das peculiaridades da situação financeira do passado (por exemplo, a inflação desenfreada).

Embora existam diversos *bankers* internacionalmente disseminados, como *Zeus* [Binsalleehet al. 2010] e *SpyEye* [Coogan 2010] – cujo roubo de credenciais bancárias de usuários está associado à modificação de arquivos e bibliotecas do sistema ou a injeção de código malicioso em processos sem apresentar referências aos bancos – os *bankers* que atingem o Brasil operam de maneira diferente. No Brasil, *bankers* costumam infectar o usuário através de *links* ou anexos em mensagens de e-mail forjadas com o remetente do banco (*phishing*), solicitando a atualização de mecanismos de segurança ou de cadastro. Para isto, os desenvolvedores de *bankers* precisam fazer o usuário crer que está realmente atendendo a um pedido de seu banco, utilizando para tanto imagens, textos, *layout* e logotipos muito próximos ou retirados do *site* original. Este tipo de *banker* tem sido visto frequentemente no ciberespaço brasileiro. Em 2006, um estudo da empresa F-Secure estimou em 30.7% a quantidade de *bankers* observados tendo por alvo bancos brasileiros [Corporation 2007]. Neste estudo, dividiu-se os *bankers* em "brasileiros" e "demais", nos quais os demais referiam-se principalmente a bancos na América do Norte, Austrália e Europa. Em 2012, um estudo da empresa Kaspersky apontou o Brasil como o país mais afetado por *bankers* [Kaspersky 2012].

Os *bankers* brasileiros atuam principalmente enganando o usuário por meio da apresentação de telas do banco alvo, aguardando assim a entrada de informações sensíveis. Isto dificulta a detecção desses *bankers*, dado que eles nada mais são do que programas com interfaces gráficas que não se baseiam em técnicas intrusivas, mas trazem embutidas em seu binário (ou realizam o *download* de) imagens obtidas dos bancos e as carregam em memória sem efetuar ações comprometedoras no registro ou no sistema de arquivos. Além disso, o fato de haver vários bancos com padrões diferentes de autenticação, necessidades de interações diversas e mecanismos de segurança distintos, faz com que a análise dinâmica em *sandboxes* (ambiente controlado passível de restauração) também seja dificultada.

Portanto, faz-se necessário o desenvolvimento de técnicas que auxiliem a análise automática e não supervisionada de *bankers*, visando aumentar sua taxa de detecção. Para isso, propõe-se a ferramenta *Esteno*, que lança mão de métodos de aprendizagem de máquina e de reconhecimento visual e textual a fim de identificar logotipos e padrões de texto relacionados a bancos brasileiros em imagens obtidas durante a execução de códigos maliciosos. Vale ressaltar que o escopo da ferramenta está restrito a análise dinâmica usando *sandbox*, logo a mesma não tem como finalidade ser usada no computador de um usuário final, como no caso dos programas antivírus, mas sim por grupos de resposta a incidentes.

Como contribuições principais deste artigo pode-se citar (i) a discussão do modo de operação dos *bankers* brasileiros, (ii) a proposta de aplicação de técnicas de análise visual e reconhecimento de textos para identificação de *bankers* e (iii) o desenvolvimento de um protótipo para testar e validar as técnicas propostas em exemplares reais vistos em atividade, mostrando resultados promissores com base em uma alta taxa de detecção. O restante do artigo está organizado como segue. Na Seção 2, o conceito de *banker* é brevemente explicado. Na Seção 3, apresenta-se a solução proposta (*Esteno*). Na Seção 4, as etapas e métodos de análise de imagens para identificar *bankers* são detalhadas. Na Seção 5, apresentam-se os testes e resultados obtidos com a solução proposta. A Seção 6 contém uma revisão da literatura associada à detecção de *bankers*. Na Seção 7, são feitas as considerações finais sobre o trabalho.

Um *banker* pode atuar de diversas maneiras, desde solicitar as informações por *e-mail* ou site clonado do banco alvo, até o *download* e execução de um programa malicioso na máquina do usuário. Este último vetor de atuação visa fazer o usuário acreditar que está acessando o ambiente real de *Internet Banking* por meio da simulação desses próprios sistemas. Assim, tais *bankers* são de difícil identificação automática por sistemas de análise dinâmica, pois além de sua principal estratégia ser a de enganar o usuário por meio de imagens e telas idênticas as dos bancos-alvo, eles possuem rotinas de execução curtas e apresentam baixa interação com o sistema operacional. Isto ocorre porque esses *bankers* consistem de apenas algumas telas com formulários a serem preenchidos com os dados bancários (agência, conta, senha de *Internet Banking*, valores da tabela de senhas etc.), os quais são enviados para o atacante via conexão com a rede.

2.1 Rotina Típica de um *Banker* Brasileiro

No Brasil, as soluções de segurança para *Internet Banking* incluem alguns fatores adicionais de autenticação, além de uma senha própria para o acesso via Internet, diferente da senha do cartão do cliente. Os fatores adicionais de autenticação são requeridos pelo banco para autorizar uma transação via Internet, como uma transferência ou um pagamento a ser efetuado.

Portanto, uma das motivações dos atacantes é capturar informações acerca dos fatores de autenticação para efetuar transações com a conta da vítima. Exemplos de fatores adicionais de autenticação são os *tokens* físicos – dispositivos de *hardware* que armazenam um certificado digital ou proveem senhas para uso único (OTP – *One-Time Password*) trocadas a cada minuto – e as tabelas de senhas – cartões com posições indexadas contendo valores que servem de senha para uma dada sessão, mas que podem ser reutilizadas em uma próxima transação caso seu índice seja requisitado. Para ilustrar o modo de atuação geral de um *banker* brasileiro, mostra-se o fluxo comum de uma rotina de execução possível na Figura 1.

No primeiro estágio, "Início", o *malware* é executado como um programa qualquer do sistema operacional, em geral sem consumir recursos excessivos. No segundo estágio é mostrada a tela inicial do banco e é apresentado algum motivo para o usuário informar seus dados. O exemplo comum é alertar o usuário sobre a necessidade urgente de atualização do mecanismo de segurança do banco. Solicita-se então a agência e a conta corrente. Esta talvez seja a parte mais importante do golpe, portanto não deve despertar suspeita, pois é a partir desta tela que o usuário será enganado ou perceberá a fraude. Devido a isto, os desenvolvedores de *bankers* em geral se utilizam de imagens originais obtidas diretamente de *sites* de *Internet Banking*.

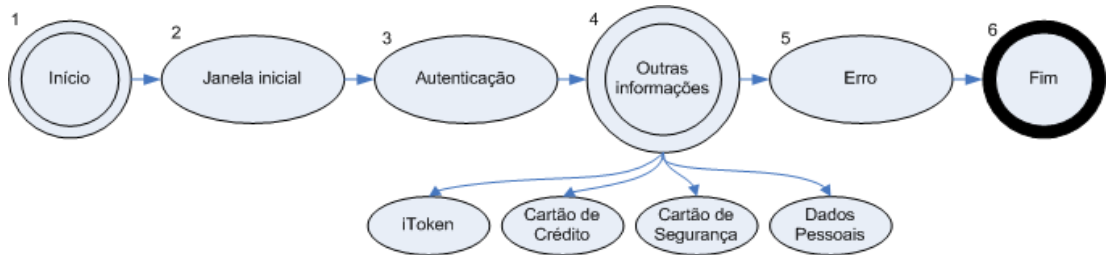


Figura 1: Rotina típica de um *banker*.

Nas janelas seguintes, as demais informações são solicitadas, sendo em geral enviadas (via métodos HTTP POST ou *e-mail*) para o atacante a cada mudança de estágio. Podem ser pedidas informações pessoais, como RG e CPF sob a alegação de atualização cadastral. Finalmente, é simulado o processamento das informações ou atualização dos mecanismos de segurança, o qual culmina em uma mensagem de erro.

Todas as janelas apresentadas durante a execução buscam possuir aparência igual ao dos sistemas legítimos dos bancos alvos, ou seja, utilizam o mesmo *brand* – logo, cores, textura, fontes e outras características que diferenciem uma marca – de forma a remeter à marca do banco, pois este é o principal fator de sucesso para conquistar a confiança do usuário sobre a legitimidade do que está sendo pedido.

Além disso, a interação do *banker* com o sistema operacional é muito semelhante a de um programa não malicioso, dado que são abertas apenas algumas janelas, o usuário preenche alguns formulários e, no final, as informações são enviadas por HTTP. Desta forma, não há nenhuma ação suspeita que possa ser utilizada facilmente para caracterizá-lo como um *malware*, somente o conteúdo das janelas e as informações pedidas. Isto é o fator de maior entrave para que a identificação dos *bankers* seja feita de modo automático por sistemas de análise dinâmica não-supervisionados e até mesmo por mecanismos antivírus.

Por fim, os textos mostrados nas telas sempre buscam usar as mesmas imagens, posicionamento de logotipos e produtos dos bancos alvos, além de termos que remetem a segurança e proteção, para dar maior credibilidade ao golpe.

3. Solução Proposta

Dadas as características levantadas sobre o comportamento apresentado por *bankers* brasileiros, fica claro que o fator mais relevante de seu sucesso são as imagens e telas mostradas às vítimas, pois são elas que persuadem o usuário a crer que está no sistema real do banco. Desta forma, a melhor maneira de identificar este tipo de *malware*, em sistemas automatizados e não-supervisionados, é analisando justamente essas imagens. Para tanto, propõe-se o *Esteno*, uma solução para a classificação automatizada de imagens e telas de *bankers*. Essas imagens são obtidas durante a execução automatizada de códigos maliciosos em um sistema de análise dinâmica de *malware* (*sandbox*), ambiente almejado para a aplicação da ferramenta.

3.1 Análise das Imagens: Logotipo, Conteúdo dos Textos e Cores

As janelas apresentadas por programas em execução nada mais são do que uma composição de vários elementos, os quais são apresentados visualmente para o usuário. No caso das janelas de *bankers*, os elementos que mais se destacam são: o logotipo do banco, o conteúdo dos textos e as cores usadas, principalmente na parte superior das imagens, que é onde se localiza o *banner* ou o cabeçalho do *site* de *Internet Banking*. Destes elementos, o logotipo é extremamente relevante, pois é o símbolo que representa o banco do cliente. Devido a isso, bem como para passar uma sensação de legitimidade e fazer com que o usuário se identifique, raramente o logotipo não está presente em uma janela apresentada por um *banker*.

Já os textos contidos nas janelas dos *bankers* costumam possuir conteúdo que remeta a segurança e proteção, como mencionado anteriormente, além de usar termos próprios e familiares de cada banco tido como alvo. Exemplos para ilustrar a afirmação anterior são os termos: "Superlinha" (Santander), "*iToken*" e o *slogan* "30 Horas" (Itaú). Com isso em mente, é possível passar as imagens por uma ferramenta de OCR (*Optical Character Recognition*) – mecanismo para converter textos presentes em imagens novamente em texto editável – para extrair partes dos textos e, posteriormente, buscar-se por termos e padrões referentes a bancos.

Quanto às cores, elas são uma das principais características representativas de uma marca (*brand*). No caso dos bancos, percebe-se que há simplicidade, pois apenas uma cor majoritária é utilizada, sendo que outra cor secundária é escolhida para a composição das bordas e *banner* das telas. Dessa forma, a identificação de uma marca

pode ser alcançada por meio do estudo da distribuição estatística das cores vermelho, verde e azul que compõem o espaço de cores RGB.

3.2 Arquitetura da Solução

A ferramenta *Esteno* foi desenvolvida na linguagem Java, por haver bibliotecas prontas capazes de codificar os requisitos delineados. Houve também o uso de ferramentas externas presentes em distribuições Linux, para facilitar alguns trabalhos, como o processamento das cores e sua distribuição.

Na Figura 2, mostra-se o esquema da arquitetura proposto para a ferramenta *Esteno*, na qual a entrada é uma imagem e a saída é a sua classificação. Internamente, existe a divisão em duas partes, a primeira abrangendo os três analisadores que extraem as características relacionadas à presença do logotipo, ao conteúdo dos textos e à estatística das cores, enquanto que a segunda é a aplicação da aprendizagem de máquina, com o algoritmo *Random Forest (RF)* (maiores detalhes na seção 5).

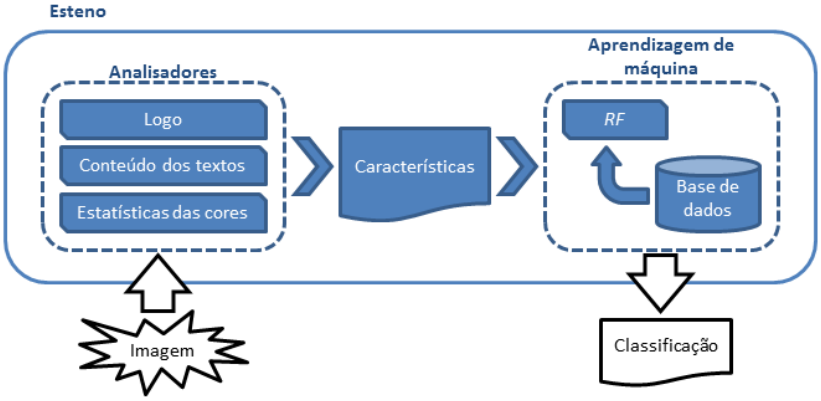


Figura 2: Arquitetura proposta para o *Esteno*.

3.3. Obtenção das Imagens

As imagens usadas no *Esteno* foram obtidas de duas maneiras: primeiramente, executou-se diversos exemplares, tirando *screenshots* das janelas ativas e associando-os ao exemplar executado. A segunda maneira deu-se pela extração de figuras encontradas embutidas dentro dos executáveis maliciosos. Para extrair as imagens presentes dentro da estrutura de dados de um arquivo binário não é necessário executá-lo, bastando-se inspecionar seu código com uma ferramenta de análise forense, como *Foremost* [for 2013]. As imagens pequenas, menores que os logotipos usados como referência, foram redimensionadas devido a limitações do algoritmo para analisa-los. Dessa forma, obteve-se uma ampla base de imagens que foram usadas no *Esteno*.

4. Classificação Visual: Técnicas Utilizadas e Detalhes de Implementação

Nesta seção serão apresentadas com maiores detalhes as técnicas utilizadas para classificação de *bankers* e como elas foram aplicadas na ferramenta *Esteno*, assim como alguns resultados individuais de cada um dos analisadores visuais implementados.

4.1 Preparação dos Analisadores

O *brand* de uma empresa, por definição, deve ser único e marcante, portanto é necessário avaliar individualmente cada banco, a fim de preparar adequadamente cada analisador, em específico, para os bancos esperados. Foi identificado que os exemplares coletados tinham por alvo os bancos: Banco do Brasil, Bradesco, Caixa Econômica Federal, Itaú e Santander; inclusive estes são os cinco maiores bancos comerciais brasileiros segundo o Banco Central do Brasil (BCB). Assim, os analisadores foram preparados com os logotipos e termos bancários utilizados por eles.

4.2 Analisador de Logotipos

Para procurar pela presença de logotipos conhecidos nas imagens, foi utilizada a biblioteca *JavaCV* [jav 2013], que encapsula uma série de bibliotecas comumente usadas em problemas de visão computacional, como é o caso da biblioteca *OpenCV*.

Na biblioteca *JavaCV* existe a classe `ObjectFinder`, uma adaptação da classe `find_obj` presente na biblioteca *OpenCV*, a qual exemplifica como buscar um objeto em uma imagem utilizando o algoritmo *Speed-Up Robust Features*, também conhecido como SURF [Bay al. 2006]. Este algoritmo é ágil e robusto para detectar características e pontos de interesse em uma imagem, independente de rotação ou escala. A classe `ObjectFinder` tem como parâmetros de entrada duas imagens, o logotipo e a imagem em que este será procurado. Inicialmente, o logotipo é processado a fim de se definir seus pontos característicos, como pode-se observar na Figura 3, representados pelos círculos vermelhos. Posteriormente, estes pontos são procurados na imagem e, se forem encontrados em uma região que condiz com o logotipo (em termos de distância e relação na posição entre os pontos), a região é delimitada por uma linha, indicando o local em que foi encontrado o objeto (logotipo).

Figura 3: Pontos característicos de três logotipos de bancos distintos.

Na Figura 4, pode-se ver três casos de funcionamento do `ObjectFinder`. No primeiro (a), foram localizados inúmeros pontos e a região do logotipo foi marcada corretamente. Cada risco indica um ponto característico encontrado e onde está presente na imagem. No segundo caso (b), foram encontrados 6 pontos e considerou-se equivocadamente que o logo está inclinado, como observamos pela delimitação, todavia a região foi marcada corretamente. Por fim, no último caso (c), foram encontrados apenas três pontos corretos, mas insuficientes para localizar o logotipo na imagem.

O fato mais importante que a Figura 4 nos permite concluir é que, independente da região delimitada, a quantidade de pontos característicos encontrados na imagem está diretamente relacionado à chance de se encontrar o logotipo na imagem. Assim, a classe `ObjectFinder` foi modificada para indicar apenas a quantidade de pontos encontrados. Cabe ressaltar que os logotipos utilizados como referência na ferramenta *Esteno* foram retirados dos *sites* originais dos bancos.

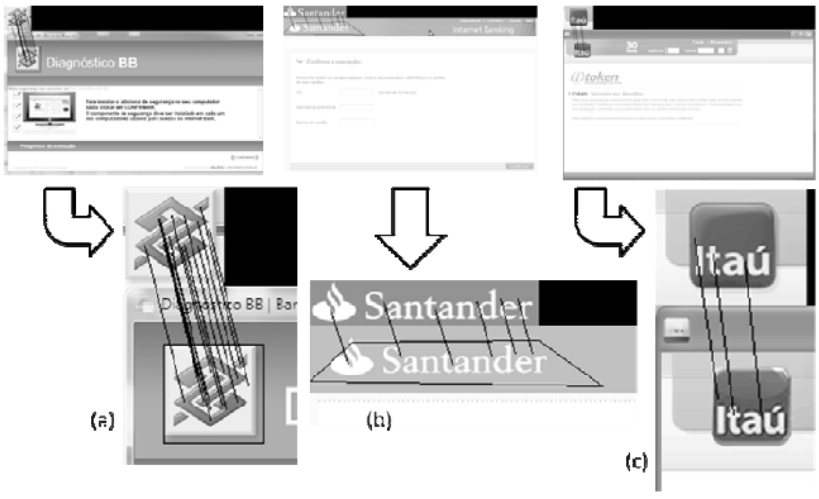


Figura 4: Três exemplos do funcionamento do `ObjectFinder`.

4.3 Analisador de Conteúdo dos Textos

Os textos presentes nas imagens foram convertidos em arquivos editáveis através da ferramenta de OCR *Tesseract* [Holley 2009] e, posteriormente, analisados quanto aos seus conteúdos. A ferramenta *Tesseract* nem sempre consegue mapear corretamente os *pixels* de uma imagem, podendo traduzir erroneamente as palavras encontradas, por não identificar ou misturar as letras. Porém, em geral observou-se que existe uma semelhança da palavra produzida com a palavra correta. Assim, para analisar o conteúdo de forma mais robusta, a qual seja capaz de tratar algumas das falhas mencionadas da *Tesseract*, foram criados padrões de expressões regulares, responsáveis por verificar a existência de nomes de bancos, produtos e termos bancários. Na Figura 5 pode-se ver alguns exemplos de expressões criadas para este fim.

```
Itaú:[EiIÍn@l][ÏtTricl][Aax][l]*[Úùúu\]]
Agência:[aA][gGql3]+[èEêÊée][nNmrv]*[cCxuUnmzl(][iIxzl]*[mnAaãà]
Banco:[BbE]an[ct][ou]
Proteção:[pP][mr][ocu]*[tîl]e[çcgq][aããāléi][uno]
Seguro:[Ssß][eêæa][gqu]u[rn,v]*[nàãaãõowzum]
```

Figura 5: Exemplos de expressões regulares utilizadas no *Esteno*.

As expressões regulares foram divididas em três grupos, sendo que cada um recebeu uma pontuação definida empiricamente após sucessivos testes. O primeiro grupo é composto pelos nomes de bancos e recebeu a pontuação "3", o segundo é formado pelos nomes de produtos e termos bancários, recebendo "2" pontos e, por último, o grupo das palavras relevantes e frequentes em *bankers*, mas que podem ser encontradas em outros programas, recebeu a pontuação "1". Na Tabela 1, apresenta-se a lista dos grupos e das palavras utilizadas no *Esteno*.

Tabela 1: Lista de grupos e palavras utilizadas como expressões regulares no *Esteno*.

Nome dos bancos (3 pontos)	Nome de produtos e termos bancários (2 pontos)	Termos relevantes e freqüentes (1 ponto)
Banco do Brasil, Bradesco, Caixa, Itaú e Santander	Superlinha, 30 Horas, iToken, Banking, Bankline, Teclado Virtual, Banco, Agência, Cartão de Débito, Cartão de Crédito e Conta Corrente	Segurança, Seguro, Proteção, Senha, Validação, Validando, Proteger, Conta e Chave

4.4 Analisador da Distribuição das Cores

No *banner* de uma tela (porção superior) há muitos elementos do *brand* representado, por exemplo, as cores, que no caso dos bancos são formadas por uma majoritária e outra secundária, conforme mencionado anteriormente. Assim, ao se analisar a distribuição das cores no espaço RGB, a cor dominante fica em destaque e contribui para a identificação de um *banker*.

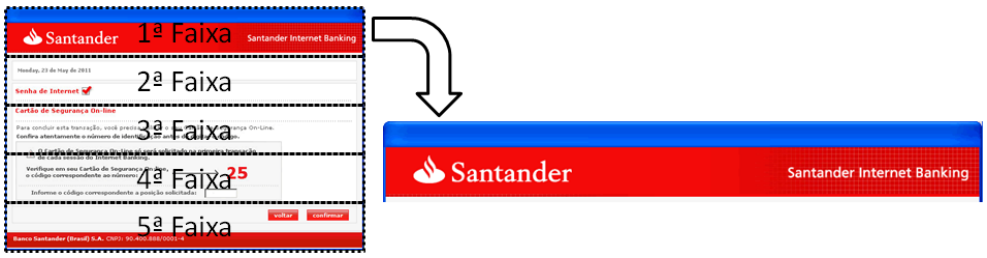


Figura 6: Divisão de faixas nas imagens para separação do *banner*.

No *Esteno*, as imagens de entrada são divididas igualmente em cinco faixas horizontais, conforme (Figura 6). A faixa mais superior, onde se localiza o *banner*, é

separada à parte para uma análise mais aprofundada. Nesta faixa, calcula-se a distribuição das cores com o pacote de ferramentas *ImageMagick* [Ima 2013].

4.5 Aprendizagem de Máquina

No *Esteno* foram definidas 27 características a serem utilizadas pela aprendizagem de máquina: seis para lidar com a presença dos logotipos (uma para cada banco e uma média), três para o conteúdo dos textos e seis para cada cor do espaço de cores RGB (somando 18). Na Tabela 2 podem-se ver todas as características adotadas.

Tabela 2: Características adotadas como atributos no *Esteno*.

Presença de logo	Conteúdo dos textos	Características estatísticas da distribuição das cores (RGB)
- Quantidade de pontos característicos: Banco do Brasil, Bradesco, Caixa Econômica Federal, Itaú e Santander. - Média dos pontos encontrados	- Quantidade de expressões regulares encontradas. - Soma da pontuação de todas as expressões encontradas. - Média de pontuação por expressão regular encontrada.	- *Red*, *Green* e *Blue*: mínimo, máximo, média, desvio padrão, curtose e assimetria.

Assim, os algoritmos *k*-NN, SVM (utilizando *kernel* Linear e RBF) e *Random Forest (RF)* do *framework* Weka [Hall et al. 2009] foram aplicados ao conjunto de testes. Estes algoritmos tiveram seus hiper parâmetros ajustados segundo a Tabela 3, com o auxílio da plataforma R Project (`www.r-project.org/`).

Tabela 3: Ajuste dos hiper parâmetros

Algoritmo	Hiper parâmetros
k-NN	k = {1, 3, 5, 11, 21, 31}
SVM Linear	C de 1e-3 a 1e4, em múltiplos de 10
SVM RBF	C e gamma de 1e-3 a 1e4, em múltiplos de 10
Random Forest (RF)	mtry = {2, 3, 5, 10, 20, 40, 60}

O *k*-NN realiza a classificação de objetos com base nos elementos de treinamentos com menor distância (calculada por uma função, por exemplo, a distância Euclidiana) no espaço de atributos [Cover and Hart 1967]. Em comparação, o SVM busca encontrar, estatisticamente, o hiperplano que melhor separe geometricamente os conjuntos [Platt 1988]. Por fim, a RF utiliza a técnica de *bagging* para combinar árvores preditoras [Liaw and Wiener 2002].

5. Testes e Resultados

Para validação da ferramenta *Esteno*, utilizou-se uma base com 1.394 exemplares† extraídos de *links* e anexos provenientes de mensagens de *phishing*, coletados entre os anos de 2010 e 2013. Os exemplares passaram pelo processo de extração de imagem explicado anteriormente, resultando em um total de 1.122 imagens.

As imagens desta base foram classificadas em dois grupos: *bankers* e *others*. O agrupamento se deu de acordo com a proposta da ferramenta, isto é, distinguir as imagens que são de *bankers* das restantes. A classe dos *bankers* teve 572 instâncias, enquanto que a classe *others*, 550. A partir destes dados, foram feitos testes no Weka, utilizando validação-cruzada (*cross-validation*) de 10-*folds* [Bengio and Grandvalet 2004] [Markatouet al. 2005], tanto para o cálculo da acurácia média, como para a escolha dos hiper parâmetros. Os melhores resultados, variando os hiper parâmetros listados na Tabela 3, dos algoritmos de aprendizagem de máquina supervisionada testados são apresentados na Tabela 4. O tempo gasto no treinamento dos algoritmos e

† Para a lista dos MD5 dos exemplares utilizados, favor contactar os autores.

na classificação de cada instância levou em média 4 minutos e menos de 1 segundo, respectivamente.

Tabela 4: Resultado dos algoritmos de aprendizagem de máquina

	k-NN	SVM Linear	SVM RBF	RF
Taxa de acerto (%)	91,5%	73,4%	91,5%	92,1%
Melhores hiper parâmetros	k = 5	C = 1	C = 1 e gamma = 10	mtry = 3

Com base na Tabela 4, o algoritmo *Random Forest* obteve o melhor resultado, gerando a classificação correta de 92,1% das instâncias, portanto foi o escolhido para compor o *Esteno*. O detalhamento do resultado obtido pela ferramenta está apresentado na matriz de confusão, típica para problemas de aprendizado de máquina, na Tabela 5.

Tabela 5: Matriz de confusão do resultado obtido pelo *Esteno*.

Classificado pelo *Esteno*			
Bankers	***Others***		
a. 537	b. 35	***Bankers***	**Classificado manualmente**
c. 54	d. 496	***Others***	

A Tabela 5 apresenta a classificação da base de imagens sobre duas perspectivas de classificação, a automatizada pela ferramenta *Esteno* (colunas) e a manual por um profissional (linhas) – revisadas por outras ferramentas, como VirusTotal (`www.virustotal.com`) e Anubis (`http://anubis.iseclab.org`). Para medir a qualidade dos resultados, observa-se na Tabela 6 as taxas de verdadeiro-positivo (VP), falso-positivo (FP), precisão, *recall* e média harmônica (*F-Measure*) – baseada na precisão e no *recall* – de cada classe definida (*Bankers* e *Others*).

Tabela 6: Valores que caracterizam a qualidade dos resultados obtidos no teste, por classe.

		VP	FP	Precisão	*Recall*	*F-Measure*
Classe	***Bankers***	93,9%	9,8%	90,5%	93,9%	92,2%
	Others	90,2%	6,1%	93,7%	90,2%	91,9%

6. Soluções existentes

Em [Buescheret al. 2011] os autores apresentam uma ferramenta de detecção de *bankers* que analisa *rootkits* de nível de usuário e detecta a instalação de *hooks* no Internet Explorer. Esses *hooks* são redirecionamentos no código que modificam o fluxo de execução do navegador para roubar informações do usuário. A ferramenta, chamada BankSafe, executa o *malware* em um ambiente controlado e utiliza assinaturas para verificar se foram feitas modificações na API usada pelo Internet Explorer. Os autores afirmam que a ferramenta possui uma taxa de detecção muito boa, mas está limitada à detecção de *malware* que utilizam *hooks*, o que não é comum no *banker* brasileiro.

Uma abordagem para detectar páginas de *phishing* na Internet é apresentada em [Medvetet al. 2008], onde os autores propõem um método de detecção visual que permite ao usuário saber de antemão se seus dados estão sendo interceptados por um atacante. A detecção se baseia em três atributos extraídos das páginas analisadas: a parte textual, as imagens e a aparência da página quando renderizada pelo navegador. Além disso, os autores propõem o uso de seu método em conjunto com outras ferramentas de detecção de *phishing* (AntiPhishand DOM AntiPhish). A sua limitação é detectar apenas tentativas de *phishing* baseadas no uso de páginas falsas carregadas pelo navegador, deixando de detectar aqueles que não utilizam o navegador para apresentar o ambiente bancário forjado. Este último caso é tratado pelo Esteno.

Outro método de detecção de *bankers* se baseia no tráfego de rede gerado pelo *malware*, ao contrário do *Esteno* que é visual. Em [Riecket al. 2010] os autores

apresentam Botzilla, uma ferramenta que utiliza assinaturas para detectar tráfego de rede característico de *malware*. O tráfego detectado está relacionado ao envio de informações para páginas controladas pelo atacante.

7. Considerações Finais

Ataques por *bankers* trazem muitos prejuízos aos usuários e instituições financeiras e, por isso, é importante a criação de medidas de proteção. Devido a natureza e modo de operação dos *bankers*, sistemas de análise dinâmica de *malware* utilizados por grupos de resposta a incidentes, podem ter dificuldades em identificá-los. Para auxiliar na detecção de *bankers* de maneira automatizada, propôs-se o *Esteno*, uma ferramenta que se utiliza de técnicas de detecção visual, classificação e identificação de padrões de texto em imagens extraídas de *malware*. Os resultados obtidos foram promissores, alcançando taxas de acerto de 92,1% na detecção de exemplares de *malware* com características de *bankers* brasileiros.

Referências

(2013). Foremost. http://foremost.sourceforge.net/.

(2013). Imagemagick.http://www.imagemagick.org/script/index.php.

(2013). Javacv.https://code.google.com/p/javacv/.

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. In *ECCV*.

Bengio, Y. and Grandvalet, Y. (2004). No unbiased estimator of the variance of K-fold cross-validation. In *J. Mach. Learn. Res.*, v. 5, pages 1089–1105. JMLR.org

Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., and Wang, L. (2010).On the Analysis of the Zeus Botnet Crimeware Toolkit. In *Privacy Security and Trust (PST), 2010 8th Annual International Conference* on, pages 31–38.

Buescher, A., Leder, F., and Siebert, T. (2011).Banksafe information stealer detection inside the web browser. In *Proceedings of the 14th international conference on RecentAdvances in Intrusion Detection, RAID'11*, pages 262–280. Springer-Verlag.

Coogan, P. (2010). Spyeye bot versus zeus bot. http://www.symantec.com/connect/ blogs/spyeye-bot-versus-zeus-bot.

Corporation, F.-S. (2007). Thetrojan money spinner. Available at http://www.f-secure.com/weblog/archives/VB2007_TheTrojanMoneySpinner.pdf.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification.*Information Theory, IEEE Transactions* on, 13(1):21–27.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor.Newsl*., 11(1):10–18.

Holley, R. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine*, 15(3/4).

Kaspersky (2012). Number of the week: 780 new malicious programs designed to steal users' online banking data detected every day. http://www.kaspersky.com/about/news/virus/2012/Number_of _the_week_780_new_malicious_programs.

Liaw A. and Wiener M. (2002).*Classification and Regression by Random Forest.* In R News 2.

Markatou, M., Tian, H., Biswas, S., and Hripcsak, G. (2005). Analysis of variance of cross-validation estimators of the generalization error. *Journal of Machine LearningResearch*.

Medvet, E., Kirda, E., and Kruegel, C. (2008).Visual-similarity-based phishing detection. In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, SecureComm '08, pages 22:1–22:6, New York, NY, USA.

Platt, J. (1998). *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press.

Rieck, K., Schwenk, G., Limmer, T., Holz, T., and Laskov, P. (2010).Botzilla: detecting the "phoning home"of malicious software. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1978–1984, New York, NY, USA. ACM.

CAFe Expresso: Comunidade Acadêmica Federada para Experimentação usando Framework Shibboleth*

Maykon Chagas de Souza[1], **Emerson Ribeiro de Mello**[1], **Michelle Silva Wangham**[2]

[1]Instituto Federal de Santa Catarina – (IFSC)

[2]Universidade do Vale do Itajaí – (UNIVALI)

maykon.c@aluno.ifsc.edu.br, mello@ifsc.edu.br, wangham@univali.br

Abstract. *To perform research in Identity Management, the researcher needs a complete infrastructure with Identity Providers (IdPs) and Service Providers (SPs) so that it can conduct your experiments. The process to provide this kind of infrastructure is time-consuming and requires a thorough knowledge on the tools, which are limiting factors for researchers who only want to conduct researches in the area. The aim of this article is to describe the CAFe Expresso wich was implemented with the purpose to facilitate the development of research on identity management and for this provides an environment for experimentation based on the Shibboleth framework, composed of IdPs, SPs, Discovery Services (DS) and the uApprove service.*

Resumo. *Para realizar pesquisa na área de Gestão de Identidades, o pesquisador necessita de uma infraestrutura completa com provedores de identidades (IdPs) e provedores de serviços (SPs) para que possa conduzir seus experimentos. O processo para disponibilizar este tipo de infraestrutura é demorado e requer um conhecimento aprofundado das ferramentas, sendo estes fatores dificultadores para pesquisadores que só desejam conduzir pesquisas na área. O objetivo deste artigo é descrever a CAFe Expresso, que foi implantada com a finalidade de disponibilizar uma federação acadêmica para experimentação baseado no framework Shibboleth composto por IdPs, SPs, Serviços de Descobertas (DS) e o serviço uApprove.*

1. Introdução

Segundo [Kallela 2008] e [Wangham et al. 2010b], o problema de gestão de identidades afeta tanto o usuário, que repete informações sem dar a devida importância ou usa senhas fracas, quanto as empresas, que além de proverem o serviço ainda precisam se preocupar com a gestão de identidades dos usuários, gerando custos administrativos e de infraestrutura. O modelo de gestão de identidades federadas surgiu como uma opção de solução para estes problemas.

No modelo de gestão de identidades federadas, objetiva-se remover a complexidade do usuário em ter que administrar um nome de usuário e senha para cada serviço que deseja acessar, permitindo que uma mesma identidade possa ser utilizada para o acesso a diferentes serviços [Jøsang e Pope 2005, Bhargav-Spantzel et al. 2007]. Uma federação

*Projeto financiado pela RNP (GId Lab).

composta por dois componentes principais: (1) provedores de identidades (*Identity Providers* – IdPs), responsáveis pela autenticação e gerenciamento das informações dos usuários de um domínio; e (2) provedores de serviços (*Service Providers* – SPs), que disponibilizam serviços para acesso dos usuários [Moreira et al. 2011].

O *framework* Shibboleth [Shibboleth 2005], desenvolvido e mantido pela Internet2[1], surgiu com o objetivo de atender federações acadêmicas, no entanto, hoje é utilizado por uma variedade de instituições em todo o mundo [Feliciano et al. 2011]. Fazendo uso das especificações SAML, o Shibboleth provê uma solução para criação de federações que oferece funcionalidades para a troca segura de dados para acessar recursos entre diferentes domínios, usufruindo do conceito de autenticação única (*Single Sign-On* – SSO).

Desde 2009, a Rede Nacional de Ensino e Pesquisa (RNP) disponibiliza o serviço da Comunidade Acadêmica Federada (CAFe[2]) às suas organizações usuárias, sendo esta construída sobre o *framework* Shibboleth. Através da CAFe, usuários de uma instituição têm acesso a serviços providos pelas demais instituições da federação, sem que para isto tenham que criar um nome de usuário e senha para cada um.

Desenvolver pesquisa aplicada na área de gestão de identidades federadas exige que os experimentos sejam conduzidos em um ambiente que implemente uma federação em sua totalidade, sendo que a complexidade de montar tal ambiente depende do *framework* escolhido [Wangham et al. 2013]. A CAFe é um ambiente de produção, ou seja, nesta federação não é permitida a realização de experimentos e assim pesquisadores que fazem prospecções tecnológicas e pesquisas científicas em gestão de identidades necessitam montar sua própria federação de testes para que possam conduzir seus experimentos.

Conceber uma federação baseada no *framework* Shibboleth para realizar experimentos práticos pode ser uma tarefa, muitas vezes, mais trabalhosa do que a implementação da pesquisa propriamente dita, o que poderia até inibir pesquisas na área. Outro fato complicador está relacionado com o custo para manter tal ambiente ativo, em termos de recursos computacionais, atualizações de segurança e de software entre outras atividades [Wangham et al. 2013].

Ciente desta necessidade e com o intuito de motivar pesquisas em Gestão de Identidades, a RNP criou em 2013 o projeto GId Lab[3] que tem como um dos objetivos disponibilizar para a comunidade acadêmica uma federação para experimentação, denominada CAFe Expresso.

O objetivo deste artigo é descrever a CAFe Expresso, uma infraestrutura voltada para pesquisadores que tenham interesse na área de Gestão de Identidades Federadas. A CAFe Expresso é constituída por provedores de identidades (IdPs), provedores de serviços (SPs) e dois diferentes serviços de descoberta, *Discovery Service* (DS): um chamado WAYF e o outro chamado Embbeded DS. Também oferece o serviço *uApprove*, que permite ao usuário saber previamente quais atributos (informações) estão sendo solicitados pelo SP que deseja acessar. Por fim, a CAFe Expresso disponibiliza um repositório com máquinas virtuais pré-configuradas, permitindo aos pesquisadores implementar uma federação completa com IdP, SP e WAYF, localmente em sua instituição, ou realizar suas

[1]http://www.internet2.edu/

[2]http://portal.rnp.br/web/servicos/cafe

[3]http://wiki.rnp.br/display/gidlab

modificações no IdP ou SP e disponibilizá-los na federação da CAFe Expresso.

Este artigo está organizado em 5 seções. Na Seção 2, são descritos os conceitos, padrões e tecnologias envolvidas na solução proposta. A Seção 3 apresenta a CAFe Expresso e os serviços utilizados para implantação do ambiente. Na Seção 4, são apresentados os trabalhos relacionados. E, por fim, a Seção 5 apresenta as conclusões.

2. Conceitos e tecnologias de Gestão de Identidades Federadas

2.1. Gestão de Identidades

A gestão de identidades pode ser entendida como o conjunto de processos e tecnologias usados para garantir a identidade de uma entidade ou de um objeto, garantir a qualidade das informações de uma identidade (identificadores, credenciais e atributos) e para prover procedimentos de autenticação, autorização e auditoria [ITU-T 2009].

De acordo com [Bhargav-Spantzel et al. 2007], um sistema de gestão de identidades é caracterizado pelos usuários, que desejam acessar algum serviço; pelas identidades, constituídas por um conjunto de atributos dos usuários, (ex.: nome, data de nascimento, CPF, RG, etc.); e por provedores de identidades e provedores de serviço.

Os modelos de gestão de identidades são classificados de acordo com a sua arquitetura. Em [Jøsang e Pope 2005, Bhargav-Spantzel et al. 2007], são descritos quatro modelos de gestão de identidades; destes, os mais utilizados são:

- Tradicional (ou isolado) – a identificação do usuário é tratada de forma isolada por cada provedor de serviços, o qual também atua como provedor de identidades (veja Figura 1a). Cabe ao usuário criar uma identidade digital para cada provedor de serviços que deseja interagir, não havendo assim o compartilhamento das identidades desses usuários entre diferentes provedores de serviços;
- Federado – provedores de identidades e provedores serviços podem estar em domínios diferentes, permitindo que usuários usem suas credenciais de um domínio para acessar serviços oferecidos em outros domínios (veja Figura 1b). Este modelo permite que os usuários possuam uma única identidade e não precisem lidar com o processo de autenticação diversas vezes, graças ao conceito de autenticação única (SSO).

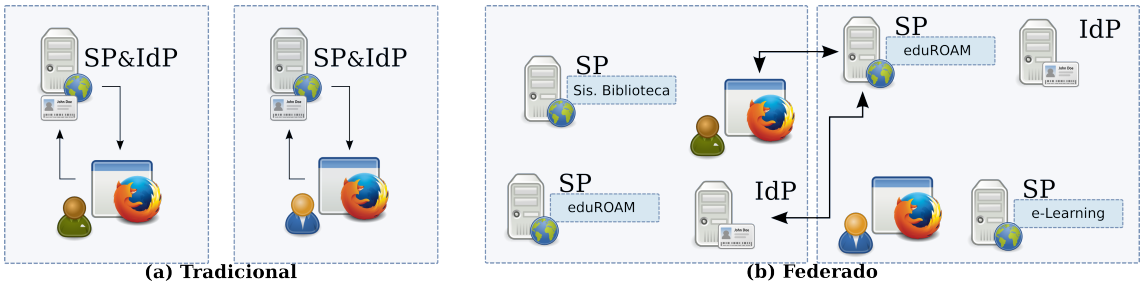


Figura 1. Modelos de Gestão de Identidades. Fonte: [Wangham et al. 2010a]

O modelo de gestão identidades federadas é uma abordagem que visa otimizar a troca de informações relacionadas a identidade por meio de relações de

confiança construídas nas federações [Camenish e Pfitzmann 2007]. Os acordos estabelecidos entre provedores de identidades e de serviços garantem que identidades emitidas em um domínio sejam reconhecidas por provedores de serviços de outros domínios [Wangham et al. 2010b].

2.2. Framework Shibboleth e CAFe da RNP

O projeto Shibboleth [Shibboleth 2005] foi uma iniciativa do consórcio americano Internet2 que teve como principal objetivo disponibilizar uma implementação de código aberto, para tratar desafios relacionados à gestão de identidades e controle de acesso em instituições acadêmicas [Wangham et al. 2010a].

Uma federação Shibboleth é composta por um grupo de organizações que usa um conjunto comum de atributos, práticas e mecanismos de segurança e permissões previamentes definidas e que permite a troca de informações e compartilhamento de serviços, possibilitando a cooperação entre membros da federação [Carmody et al. 2005].

O *framework* está fundamentado sobre padrões abertos como o *eXtensible Markup Language* (XML) e *Security Assertion Markup Language* (SAML) e provê uma forma fácil para que aplicações *web* usufruam das facilidades providas pelo modelo de identidades federadas [Wangham et al. 2010b].

O *framework* Shibboleth é composto por: (1) IdP que é a entidade responsável pelo gerenciamento das identidades dos usuários, seus atributos, gerenciamento da autenticação e declarações de atributos, e (2) SP, entidade responsável pelo gerenciamento de segurança dos serviços disponibilizados que, com base nas declarações de atributos recebidas do IdP, permite o acesso a estes serviços. As informações do usuário são trocadas entre SP e IdP através de um contexto de segurança, estabelecido após a autenticação do usuário. Isso é possível devido à relação de confiança existente entre SP e IdP [Kallela 2008].

A CAFe utiliza o *framework* Shibboleth e tem como objetivo congregar todas as universidades e instituições de pesquisa brasileiras. A metodologia adotada para construção da infraestrutura básica da CAFe consiste na utilização de padrões e soluções de *softwares* já disponíveis e adotados por outras federações e da implementação e experimentação de ferramentas auxiliares para apoiar a implantação de provedores de identidades e de serviços [Moreira et al. 2011].

2.2.1. uApprove

Desenvolvido pela SWITCH[4], o *uApprove* é uma extensão para o IdP Shibboleth que possibilita ao usuário conhecer previamente quais atributos estão sendo solicitados pelo SP que o usuário deseja acessar, dando ao usuário a opção de autorizar ou não a liberação de seus atributos para o SP em questão. Além disto, o uApprove oferece uma garantia aos provedores de identidades relacionada com o consentimento de seus usuários sobre a liberação de seus atributos para o SP. Isto é feito através da apresentação de um termo de uso do serviço no primeiro acesso do usuário. Se o usuário concordar com o termo e se

[4]http://www.switch.ch/

este não for abusivo ou ferir qualquer lei, então o administrador do IdP estaria protegido contra futuras queixas de seus usuários.

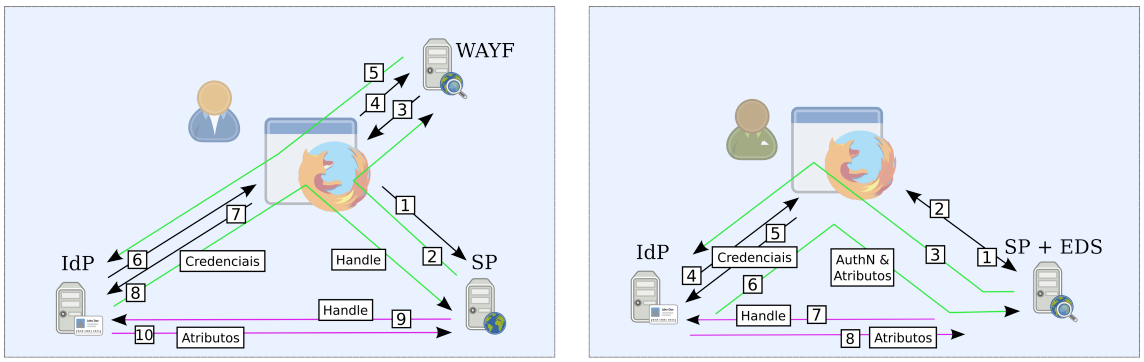
O *uApprove* não permite aos usuários escolher quais atributos serão liberados para o SP que este está tentando acessar. O usuário aceita liberar todos os atributos ou nenhum destes atributos. Neste caso, o seu acesso ao serviço poderá ser negado pelo provedor de serviço. O mesmo vale sobre o termo de uso de serviço apresentado pelo IdP ao usuário no primeiro acesso. Se o usuário não concordar, então não poderá usufruir do serviço oferecido pelo IdP.

2.2.2. WAYF e Embbeded DS

O padrão SAML possui um protocolo para descoberta de serviços, o *Discovery Service* (DS), que possibilita a descoberta de provedores de serviços e de identidades [OASIS 2008]. Com o *framework* Shibboleth é possível fazer uso de dois serviços de descoberta de provedores de identidades: *Where Are You From* (WAYF)[5]; e *Embbeded Discovery Service* (EDS)[6]. O funcionamento básico dos dois é semelhante: apresentar para o usuário uma lista com todos os provedores de identidades da federação e, uma vez que o usuário escolhe um provedor nesta lista, redirecioná-lo para a página do provedor de identidades escolhido.

A diferença entre os dois está na forma que se apresentam para o usuário. Enquanto o WAYF tem uma página *web* própria, em uma URL distinta e explícita para o usuário, o EDS aparece embarcado na página do provedor de serviço, dando a impressão para o usuário que se trata de um componente do próprio provedor de serviços e não uma outra entidade.

O WAYF consiste de um serviço, disponível em um provedor de serviços, que mantém uma base dos metadados de todos os IdPs e SPs da federação, responsável assim pelo estabelecimento das relações de confiança entre os membros [Shibboleth 2005, Kallela 2008, Wangham et al. 2010b]. O EDS faz uso da mesma base mantida pelo WAYF, porém com uma forma de apresentação diferente.



a) WAYF (tradicional) b) Embbeded DS

Figura 2. Fluxo de mensagens realizadas entre usuário ao utilizar WAYF ou EDS

[5]https://wayf.switch.ch/

[6]http://shibboleth.net/products/embedded-discovery-service.html

A Figura 2a apresenta as trocas de mensagens entre o navegador do usuário, o SP, o WAYF e entre o IdP. Neste caso, o usuário ao acessar a página do SP (passo 1) é redirecionado à página do WAYF (passo 2). O usuário escolhe seu IdP (passos 3 e 4) e é redirecionado à página do seu IdP (passo 5). Por fim, após passar pelo processo de autenticação (passos 6 e 7), é finalmente redirecionado à página do SP (passo 8). Se possuir os atributos exigidos pelo SP, o usuário terá acesso ao recurso.

A Figura 2b ilustra as trocas de mensagens fazendo uso do EDS. O usuário acessa a página do SP (passo 1) e nesta mesma página, através do EDS, indica qual é o seu IdP (passo 2). Neste momento, o usuário é redirecionado à página do IdP (passo 3) e, após passar pelo processo de autenticação (passos 4 e 5), finalmente, é encaminhado para o serviço desejado (passo 6).

3. CAFe Expresso

Mantido pela RNP como plataforma de apoio aos pesquisadores brasileiros, principalmente os participantes do Programa de Gestão de Identidades (PGID) e dos Grupos de Trabalhos (GTs) da RNP, o projeto GId Lab provê uma Infraestrutura de Autenticação e Autorização (IAA). Um de seus objetivos consiste na oferta de uma federação Shibboleth para experimentos, permitindo aos pesquisadores de qualquer instituição de ensino do Brasil, desenvolver serviços ou disponibilizar um provedor de identidades.

O projeto GId Lab provê ainda o SGCI[7] da ICPEdu[8], um software desenvolvido para o âmbito acadêmico que permite fazer a implantação, gerenciamento de uma Infraestrutura de Chave Pública (ICP) e para emissão de certificados digitais [Wangham et al. 2013].

A CAFe Expresso é um dos serviços do GId Lab e é composta por: três IdPs com usuários com diferentes perfis e atributos; três SPs configurados para proteger aplicações *web* em PHP, Java e Python. Assim, pesquisadores que queiram realizar experimentos com serviços, poderiam disponibilizá-los em um destes três SPs, de acordo com a linguagem de programação escolhida. Estes provedores de serviços e de identidades estão espalhados pelos Pontos de Presença (PoPs) da Rede Ipê[9] da RNP.

Em um dos IdPs da CAFe Expresso foi integrado o *uApprove*, módulo de consentimento do usuário sobre a liberação de atributos (veja Seção 2.2.1). Também são oferecidas as duas formas para o serviço de descoberta: o WAYF é oferecido em uma máquina virtual isolada e específica para isto; e o EDS foi integrado a um dos SPs da federação.

Por fim, a CAFe Expresso provê também um repositório com máquinas virtuais pré-configuradas, permitindo aos pesquisadores criarem uma federação local completa, com IdP, SP e WAYF, ou ainda disponibilizarem seus IdPs ou SPs na própria CAFe Expresso. A Figura 3 ilustra todos os componentes oferecidos pela CAFe Expresso.

3.1. Infraestrutura e Serviços da CAFe Expresso

A infraestrutura da CAFe Expresso é composta por 8 máquinas virtuais com Ubuntu Linux, espalhadas geograficamente pelos Pontos de Presença (PoPs) da RNP, sendo estes;

[7]https://projetos.labsec.ufsc.br/sgci

[8]http://www.rnp.br/servicos/icpedu.html

[9]http://www.rnp.br/ipe/

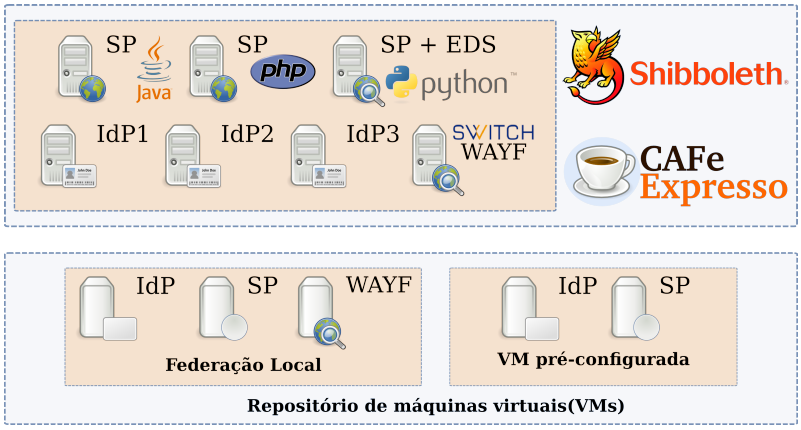


Figura 3. Infraestrutura disponível na CAFe Expresso

Mato Grosso, Mato Grosso do Sul, Goiás, Piauí, Maranhão e Alagoas. Para garantir a segurança dos servidores, foi definida uma política de segurança para proteger o serviço de administração remota (SSH) e o servidor *web* (Apache HTTP e Apache-Tomcat) implementada via filtro de pacotes (iptables). Algumas configurações de segurança nos próprios serviços oferecidos foram habilitadas, como o monitoramento dos registros de acessos *logs* do sistema e das aplicações.

Foram desenvolvidos roteiros de configuração e implantação de todo o ambiente com o objetivo de facilitar a implantação de uma federação com o *framework* Shibboleth. Estes roteiros estão disponíveis na página wiki do projeto GIDLab[10].

3.2. Operação da CAFe Expresso

Para que os pesquisadores possam fazer uso dos ambientes disponibilizados, é necessário o preenchimento de um formulário[11] de solicitação, onde o interessado deve informar qual o propósito de uso, detalhes sobre o projeto, que tipo de infraestrutura utilizará, previsão do tempo de uso do ambiente, dados dos responsáveis e contatos técnicos. Após o cadastro, o pesquisador é contatado e recebe orientações de como proceder para realizar o *download* das VMs e as orientações de configurações das mesmas, ou se for caso, como disponibilizar sua aplicação em um dos provedores de serviço padrões da CAFe Expresso.

Desde julho de 2013, a CAFe Expresso atende pesquisadores do Brasil, que fazem uso de máquinas virtuais pré-configuradas ou da infraestrutura de federação local para desenvolver suas pesquisas em gestão de identidades. Até o momento, 10 projetos de pesquisas já fizeram uso das facilidades providas pela federação para experimentação, sendo que 4 ainda estão em andamento. Dentre estes, pesquisadores das instituições UFF, UFRN, UFPE, UFMA, UFRGS, PUC-Rio e UFSC utilizaram o ambiente desenvolvido na CAFe Expresso.

As pesquisas tratam de desenvolvimento de serviços como: transposição de credenciais para testbeds da Internet do Futuro, projeto este em conjunto com federações Européias; um módulo *web* de visualização de dados coletados de redes sem fio, usando identidades federadas; testes de autenticação federada com OpenStack; análise

[10]https://wiki.rnp.br/display/gidlab/Procedimentos+operacionais+da+CAFe+Expresso
[11]http://bit.ly/formularioCadastroGIdLab

de integração de ambientes em nuvem (*cloud*) privada, usando Shibboleth e OpenID; interoperabilidade entre Shibboleth e OpenAM; infraestrutura de controle de acesso baseado em políticas; implantação de IdPs para o projeto de Computação Em Nuvem Para Ciência (CENPC) do GT-CNC da RNP, entre outros.

3.3. Pesquisa de Satisfação de Uso

Para avaliar as funcionalidades oferecidas na CAFe Expresso, assim como suas deficiências, sugestões de melhorias e verificar a importância de um ambiente para experimentação, foi aplicado uma pesquisa de uso. Os avaliadores convidados foram membros do Comitê Técnico de Gestão de Identidades (CT-GId) da RNP e 4 alunos do mestrado de Computação Aplicada da UNIVALI. A pesquisa foi realizada no período de 19 à 29 de junho de 2014, obtendo um total de 19 respostas de um grupo de 37 pessoas.

A pesquisa foi dividida em três partes. Na primeira parte, os entrevistados avaliaram o acesso federado usando o *uApprove*, seguindo um roteiro de experimento. Na segunda parte, os entrevistados avaliaram o acesso federado e as funcionalidades providas pelo EDS, seguindo um segundo roteiro de experimento. Por fim, na terceira parte os entrevistados avaliaram a CAFe Expresso como um todo, os roteiros, a linguagem utilizada, se as mensagens de erro (quando aparecem) são claras e se o uso do ambiente foi satisfatério. A pesquisa foi composta com perguntas objetivas (obrigatórias) e descritivas (não obrigatórias); neste último caso, são solicitações de sugestáes, ou descriçoes de problemas encontrados, caso ocorressem.

Dos entrevistados, somente 16%, nunca havia acessado um provedor de identidades da CAFe. 16% avaliaram seu nível de conhecimento sobre autenticação federada como baixo; outros 16% informaram conhecimento razoável; 53% dos entrevistados indicaram ter nível de conhecimento bom sobre autenticação federada e o Shibboleth. Já 16% dos entrevistados se consideram experientes.

Com relação a problemas para realizar as atividades, 16% dos entrevistados indicaram que enfrentaram problemas no acesso ao SP java. Isto ocorreu devido a uma falha de disponibilidade no PoP-MS, o qual hospedava a máquina.

O *uApprove* possui duas funcionalidades: apresentação do Termo de Uso do IdP, que informa ao usuário quais sãos seus direitos e deveres ao utilizar aquele IdP, e apresentação dos atributos liberados pelo IdP para o SP. Ambas as funcionalidades foram apresentadas para todos os entrevistados, antes de realizarem o experimento. Assim, 84% dos entrevistados indicaram que o EDS melhora a usabilidade e, ao comparar as funcionalidades do WAYF e EDS, 79% dos entrevistados responderam que o EDS facilita a escolha do IdP.

Foi possível constatar que as funcionalidades disponibilizadas pela CAFe Expresso e que não são encontradas na CAFe facilitam o entendimento do processo de autenticação e melhoram a usabilidade da federação. Além disso, a disponibilidade de SPs com serviços diferentes do serviço de homologação de atributos colaborariam para o entendimento do funcionamento do ambiente federado.

4. Trabalhos relacionados

O Fed-lab[12] é um ambiente para *testbeds* de federação usando SAML, desenvolvido pela GÈANT, uma rede pan-Européia de Educação e Pesquisa, que interliga redes de ensino e pesquisa nacionais (NREN). O Fed-lab possibilita a criação de SPs e disponibiliza IdPs para autenticação, todos baseados no *framework* SimpleSAMLphp[13]. O pesquisador pode realizar a configuração de um SP com informações próprias para depois colocar sua aplicação PHP no SimpleSAMLphp SP criado. O Fed-lab disponibiliza SimpleSAMLphp IdPs para registro de usuários para testes no ambiente.

TestShib[14] é um ambiente para testar a instalação do Shibboleth desenvolvido pela Internet2 e atualmente mantido por uma comunidade de pesquisadores de gestão de identidades federadas. Neste ambiente, o pesquisador pode realizar os testes para validação de um SP ou um IdP Shibboleth pós-instalação. O pesquisador configura as informações do seu provedor de acordo com as orientações na página do TestShib e realiza os testes, que ao término informarão ao pesquisador se sua instalação está funcional ou não.

O Fed-lab provê um ambiente federado para testes, porém não disponibiliza modos de configurar um IdP, somente SPs. Desta forma, pesquisadores não poderão realizar experimentos com o IdP. O TestShib disponibiliza uma forma de testar a implantação de IdP ou SP, providos pelos pesquisadores. Porém, o pesquisador ainda terá de despender tempo e conhecimento para configurar o provedor que deseja testar. O serviço oferecido pela CAFe Expresso seria equivalente a combinação dos serviços providos pelo Fed-Lab e TesShib e ainda permite a pesquisa com provedores de identidades, uso do uApprove e EDS e oferta de imagens de máquinas virtuais pré-configuradas, prontas para o uso.

5. Conclusões

Gestão de identidades federadas é uma área ativa de pesquisa, sendo que muitos trabalhos desenvolvidos nesta área precisam realizar experimentos com soluções e *frameworks* consolidados como o Shibboleth. Desenvolver pesquisas aplicadas na área de gestão de identidades federadas exige que os experimentos sejam conduzidos em um ambiente que implemente uma federação em sua totalidade. Configurar uma federação para realizar experimentos de uma pesquisa, pode ser uma tarefa mais árdua e demorada do que a implementação da pesquisa propriamente dita [Wangham et al. 2013].

Algumas sugestões para continuidade do trabalho descrito neste artigo são: (1) a implantação do IdP+, que é um IdP para tradução de credenciais de segurança, que permite a geração de certificados X.509 para que aplicações não *web* possam fazer uso de autenticação federada Shibboleth, (2) implantação do Serviço Gerador de Certificados (SGC) que permite a tradução de credenciais Shibboleth em certificados digitais, e que sejam consumidas por serviços que requerem estes tipos de certificados, (3) integração entre a federação CAFe Expresso, que utiliza o padrão SAML através do *framework* Shibboleth e outras tecnologias de gestão de identidades federadas, como OAuth[15] e OpenID Connect[16] que implementam outros padrões de comunicação, diferentes do SAML.

[12]https://fed-lab.org/
[13]https://simplesamlphp.org/
[14]http://www.testshib.org/
[15]http://oauth.net/
[16]http://openid.net/

Referências

Bhargav-Spantzel, A., Camenish, J., Gross, T., e Sommer, D. (2007). User centric: A taxonomy and open issues. pages 493–527.

Camenish, J. e Pfitzmann, B. (2007). *Security, Privacy and Trust in Modern Data Management*, chapter Federated Identity Management, pages 213–238.

Carmody, S., Erdos, M., Hazelton, K., Hoehm, W., Morgan, B., Scavo, T., e Wasley, D. (2005). *InCommon Technical Requirements and Information*.

Feliciano, G., Agostinho, L., Guimarães, E., e Cardozo, E. (2011). Gerência de identidades federadas em nuvens: Enfoque na utilização de soluções abertas. In *Minicursos do XI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC)*, pages 182–231. Sociedade Brasileira de Computação (SBC).

ITU-T (2009). *NGN Identity Management Framework – Recommendation Y.2720*.

Jøsang, A. e Pope, S. (2005). User centric identity management. *AusCERT Asia Pacific Information Technology Security Conference*.

Kallela, J. (2008). Federated identity management solutions. *Seminar on Internetworking, TKK T110.5190*.

Moreira, E. Q., Foscarini, A. D., Junior, G. C. S., Alixandrina, L. A. O., Neto, L. P. V., e Rosseto, S. (2011). *Federação CAFe: Implantação do Provedor de Identidade*. Rede Nacional de Ensino e Pesquisa (RNP), Rio de Janeiro.

OASIS (2008). Security Assertion Markup Language (SAML). Technical Report Technical Overview, OASIS.

Shibboleth (2005). Shibboleth architecture. Technical report, Internet2.

Wangham, M. S., Mello, E. R., Böger, D. S., Fraga, J. S., e Guerios, M. (2010a). Uma infraestrutura para tradução de credenciais de autenticação para federações shibboleth. In *X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 360–447. Sociedade Brasileira de Computação (SBC).

Wangham, M. S., Mello, E. R., Böger, D. S., Guerios, M., e Fraga, J. S. (2010b). Gerenciamento de identidades federadas. In Barreto, L. P., editor, *Minicursos do Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais*, pages 3–52. Sociedade Brasileira de Computação (SBC).

Wangham, M. S., Mello, E. R., Souza, M. C., e Coelho, H. (2013). Gidlab: Laboratório de experimentação em gestão de identidades. In *Anais XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 481–486. Sociedade Brasileira de Computação (SBC).

Estudo e Análise de Vulnerabilidades Web

Wagner Aparecido Monteverde, Rodrigo Campiolo

[1]Departamento Acadêmico de Computação
Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão - PR

wagnermonteverde@outlook.com, rcampiolo@utfpr.edu.br

Abstract. *Web security is important to provide protection to clients and services in Web. Several Web vulnerabilities are exploited every day and the attacks have increased due to new tools and Web applications. In this work is carried out an analysis of Web vulnerabilities in different kinds of applications. A set of heterogeneous and brazilian Web sites was selected and analysed by open source tools. Consequently, the main forms of attacks used in Web applications have been investigated. Our results show how Web vulnerabilities can be exploited easily. So, it is verified that websites need to improve their security urgently.*

Resumo. *A segurança em aplicações Web é importante para prover a proteção aos clientes e serviços na Web. Inúmeras vulnerabilidades Web são exploradas a cada dia e os ataques tem se tornado mais frequentes devido a facilidade introduzida por ferramentas de exploração e pelo aumento de aplicações e o uso da Web. Neste trabalho é realizado o estudo e a análise de vulnerabilidades em diferentes tipos de aplicações Web. São usadas ferramentas para identificação de vulnerabilidades em uma amostra de sítios Web heterogêneos e brasileiros. Por consequência, foram investigadas as principais formas de ataques utilizadas em aplicações Web. Os resultados indicam a necessidade urgente de melhorias na segurança, principalmente em sítios Web menores e regionais.*

1. Introdução

No início, a Web foi criada sem grandes preocupações com segurança, tendo como objetivo principal a disponibilização de conteúdos adequados aos recursos disponíveis na época. Mas, a cada dia que passa a Web vem se consolidando como um dos principais meios de comunicação, de relacionamentos e de negócios. De uma maneira crescente empresas disponibilizam informações, produtos, serviços e, ainda, realizam negócios cada vez mais importantes. Diante desses fatos tornou-se de extrema necessidade garantir a segurança nas aplicações Web. Por se tratar de serviços remotos, executado muitas vezes distribuidamente, conceitos fundamentais de segurança como confidencialidade, integridade, disponibilidade e autenticidade devem estar presentes nos sistemas para diminuir os riscos de ataques devido à exploração de vulnerabilidades [Mello et al. 2006].

Vulnerabilidades são condições que quando exploradas por pessoas mal intencionadas podem resultar em falhas de segurança [Shirey 2000]. As vulnerabilidades Web são o resultado de um conjunto de fatores que, em sua maioria, envolve todo processo de desenvolvimento, de manutenção e de uso das aplicações. Prazos curtos de entrega das aplicações aliados a processos falhos de desenvolvimento resultam em maior taxa de

falhas de segurança nas aplicações [CERT.BR and CGI.BR 2012]. Um ponto importante também é a qualificação técnica dos desenvolvedores e uma política de revisão constante e de melhoria contínua, que se não aplicadas corretamente podem levar a possíveis vulnerabilidades. Mesmo a configuração dos servidores e equipamentos de rede, quando feitas de maneira padrão, podem oportunizar brechas graves de segurança independentemente da qualidade e dos padrões de segurança aplicadas no processo de desenvolvimento [Bishop and Frincke 2005].

O estudo e a análise das principais vulnerabilidades Web possibilita aos desenvolvedores uma compreensão dos principais erros cometidos no ciclo de desenvolvimento de software, evitando assim, custos futuros com manutenção relacionadas a falhas de segurança. Neste trabalho é realizado a identificação e a exploração de vulnerabilidades Web por meio de ataques executados em cenários reais, enumerando assim as vulnerabilidades encontradas e o impacto que cada uma representa ao sítio Web e o seu respectivo negócio.

2. Trabalhos Relacionados

Falhas de segurança em aplicações Web tem sua origem principalmente devido ao não tratamento das entradas de informações nas aplicações. [Halfond et al. 2006] apresentam uma revisão sobre os diferentes tipos de ataques de Injeção de SQL conhecidos e, para cada tipo de ataque, são descritos exemplos de como podem ser efetuados. Também são abordadas as formas de combater e prevenir cada um desses ataques.

[Dacosta et al. 2012], [Huluka and Popov 2012] e [Kolšek 2002] abordam especificamente as falhas decorrentes da Quebra de Gerenciamento de Sessão, bem como os métodos utilizados por aplicações Web para criar um ambiente de autenticação que facilite a interação do usuário, como por meio do armazenamento de *cookies*, que são um dos principais alvos de ataque nesse tipo de vulnerabilidade.

[Vogt et al. 2007] destacam os principais conceitos da vulnerabilidade conhecida como Scripts entre Sites (XSS) e sua finalidade de roubo e envio de informações sigilosas para terceiros. [Pelizzi and Sekar 2012] enfatizam o uso de ferramentas de detecção do XSS no lado do cliente, como extensões utilizadas nos navegadores Web. Destacam ainda os pontos positivos e negativos dessa medida contra XSS, avaliando assim tipos de filtros e a usabilidade das extensões como ferramentas de proteção.

[Hinrichs et al. 2013] apresentam uma linguagem declarativa para desenvolvimento de aplicações Web elaborada para eliminar automaticamente várias vulnerabilidades comuns. Entre as vulnerabilidades evitadas com o uso da linguagem estão a Injeção de SQL o XSS e a Falta de Controle em Nível de Acesso. Hinrichs ainda destaca os pontos negativos da linguagem em não prevenir contra outras vulnerabilidades comuns, como a Falsificação de Solicitação entre Sites (CSRF) e a Quebra de Gerenciamento de Sessão.

[Koshutanski and Massacci 2008] propõem um novo modelo de controle de acesso interativo, onde os servidores e aplicações devem interagir com os clientes por meio da solicitação de credenciais para garantir o acesso ou negá-lo. Esse controle de acesso baseia-se em um modelo formal criado a partir de políticas específicas que identifica os serviços de raciocínio de dedução e de consistência para implementação de sistemas autônomos.

[Lin et al. 2009] exploram a Falsificação de Solicitação Entre Sites (CSRF) ilustrando ataques em ambientes reais, e exemplificando os modelos de ameaças de CSRF, fornecendo assim uma compreensão detalhada da vulnerabilidade e ilustrando ainda os recursos de proteção e segurança em cenários reais.

[Wu et al. 2010] ressaltam que repositórios de software são ricas fontes de informação sobre as mais diversas vulnerabilidades que ocorrem no ciclo de desenvolvimento de software. Destacam que essas informações são replicadas por diversas bases de dados. Uma única vulnerabilidade em um repositório de software pode se estender por vários componentes e ter interações multidimensionais com outras vulnerabilidades, tornando assim os componentes facilmente exploráveis.

Apesar de todos os trabalhos abordarem possíveis soluções diferentes para vulnerabilidades distintas e, até mesmo, a criação de uma nova linguagem de programação segura, a maneira mais eficiente ainda é seguir as práticas de programação segura durante o processo de desenvolvimento de aplicações Web. O presente trabalho apresenta o estudo e exploração das principais vulnerabilidades Web em um conjunto variado de sítios para apresentar a simplicidade de exploração e a urgência de melhorias no processo de desenvolvimeto de software seguro dentro das empresas.

3. Métodos de Pesquisa

Na presente pesquisa foram analisados sítios Web de vários segmentos incluindo sítios de domínio nacional e regional, desde de lojas virtuais até sítios informativos de pequenas empresas ou órgãos públicos e também uma rede social conhecida. Considerou-se então primeiramente sítios que utilizaram arcabouços em seu desenvolvimento, bem como sítios que tenham em sua base *Content Management System* (CMS) como gerenciadores de conteúdo. O uso de arcabouços no desenvolvimento pode induzir os programadores a falhas de configuração devido a falta de conhecimento técnico ou mesmo por falhas no próprio arcabouço utilizado.

O foco das buscas de vulnerabilidades foi orientado pelo relatório *Owasp Top Ten* [Ten 2013]. Este relatório é desenvolvido por uma fundação que atualmente é referência em segurança Web, a OWASP, que é referenciada inclusive por normas como a *Payment Card Industry* PCI [Virtue 2008]. O *Top Ten* contempla as 10 principais vulnerabilidades encontradas em sistemas Web nos últimos 3 anos a nível mundial e estão dispostas em categorias entre A1 até A10.

As ferramentas utilizadas para buscar vulnerabilidades foram os *Web Scanners*. Tais ferramentas automatizam o processo de busca por falhas, sendo que alguns permitem a configuração do perfil das vulnerabilidades desejadas. O ponto positivo da utilização dos *Scanners* foi a automatização do processo de busca, em contrapartida podem ocorrer falsos positivos, ou seja, a ferramenta pode indicar uma vulnerabilidade inexistente devido ao comportamento inesperado da aplicação. Os principais *Web Scanners* utilizados neste estudo foram o *W3af* [W3af 2013] e o *VEGA* [SUBGRAPH 2013].

Os *Web Scanners W3af* e *VEGA* possibilitaram a identificação das vulnerabilidades existentes nos sítios Web bem como as *URLs* específicas das falhas e os parâmetros vulneráveis nas aplicações. Foram efetuadas requisições HTTP para as essas *URLs* e, com a ajuda da ferramenta *Burp Suite* [Portswigger 2014] configurada como um *proxy* local,

foi realizada a captura das requisições com todas as informações da mesma, como por exemplo, parâmetros e *cookies*. Essas informações das requisições HTTP foram salvas em modo texto para utilização posterior como fonte de informação para outras ferramentas de exploração.

A exploração de algumas vulnerabilidades Web ocorreu por meio de arcabouços específicos. No entanto, para outras, foi utilizado um conjunto diversificado de ferramentas. Dentre os arcabouços utilizados para a exploração, o SQLMap [G. and Stampar 2013] foi usado para a exploração de Injeção de SQL. As informações das requisições HTTP salvas anteriormente foram utilizadas pelo SQLMap e possibilitaram a extração de dados sensíveis das aplicações vulneráveis.

Ferramentas de análise de rede, como o *Wireshark* [Orebaugh et al. 2006], foram utilizadas para visualizar o tráfego de pacotes nas redes exploradas, possibilitando a filtragem de pacotes e a análise de seu conteúdo. Em um ataque específico, o *wireshark* foi utilizado para extrair um *cookie* de um pacote capturado na rede, que posteriormente foi injetado em um navegador para fixar uma sessão ativa de uma aplicação Web.

Além dos ataques específicos a aplicações Web, também foi executado um ataque *offline* a um *hash* de uma senha extraída de uma base de dados remota. Para tal, foi utilizada a ferramenta *John The Ripper* [Anderson 2014]. No ataque foi utilizada a versão do *John The Ripper* com suporte a Interface de Troca de Mensagens - MPI [Squyres and Lumsdaine 2004], que possibilitou o processamento do *hash* em várias *threads*.

4. Experimentos e Resultados

Os experimentos de exploração de vulnerabilidades foram executados em cenários reais. As vulnerabilidades encontradas durante as varreduras utilizando os *Web Scanners* foram reportadas aos responsáveis pelos sítios Web e, com a devida autorização dos mesmos, foram executadas a exploração de algumas vulnerabilidades encontradas. Na seção 4.1 são descritos os resultados das varreduras nos sítios Web e nas seções 4.2 a 4.6 os resultados de explorações de vulnerabilidades específicas.

4.1. Execução de varreduras nos sítios Web

Para a execução das varreduras foram escolhidas categorias específicas de sítios Web, de forma que se pudesse abranger uma maior diversidade de sítios com um número reduzido de amostras de cada categoria. Inicialmente foram estabelecidas as seguintes categorias de sítios Web para análise: comércio eletrônico, religiosos, acadêmicos, grandes portais, sítios que utilizam CMSs, governamentais, regionais e de conteúdo adulto. A visão geral das varreduras efetuadas pode ser observada na Tabela 1 que apresenta os tipos de sítios o tempo médio gasto por sítio e a quantidade de sítios por categoria.

Tabela 1. Visão geral das varreduras.

Tipos de Sítios Web	Tempo Médio de Varredura	Quantidade de Sítios Web
Comércio Eletrônico	6 horas	4
Religiosos	30 minutos	1
Acadêmicos	3 horas	2
Grandes Portais	6.3 horas	1
Sítios que Utilizam CMS	56 minutos	2
Governamentais	30 minutos	1
Regionais	40 minutos	4
Conteúdo Adulto	1 hora	1

A visão detalhada das varreduras pode ser vista na Tabela 2, contendo o total de vulnerabilidade por sítio Web classificado pelo grau de risco disponibilizado pela OWASP [Ten 2013]. Os sítios são numerados de 1 a 16, especificando o tipo de sítio e a severidade das vulnerabilidades encontradas no mesmo.

Tabela 2. Distribuição dos Sítios e as Vulnerabilidades Encontradas.

Sítios/Tipo de Sítios	Vuln. Severa	Vuln. Moderada
Sítio 1 / Regional	0	1
Sítio 2 / Comércio Eletrônico	0	2
Sítio 3 / Regional	1	2
Sítio 4 / Grande Portal	0	1
Sítio 5 / Regional	0	2
Sítio 6 / Comércio Eletrônico	1	2
Sítio 7 / Comércio Eletrônico	1	1
Sítio 8 / Acadêmico	1	2
Sítio 9 / CMS	1	2
Sítio 10 / Religioso	1	2
Sítio 11 / Acadêmico	1	1
Sítio 12 / Governamental	1	1
Sítio 13 / Regional	1	1
Sítio 14 / CMS	0	1
Sítio 15 / Comércio Eletrônico	1	1
Sítio 16 / Conteúdo Adulto	1	2

Ao todo foram investigados 16 sítios Web, e em todos os sítios analisados foram encontradas vulnerabilidades abordadas no OWASP Top 10, sendo que 33% das vulnerabilidades foram classificadas como severas pelo grau de classificação de risco da OWASP. O nível de dificuldade de exploração dessas vulnerabilidades é considerado como "fácil" pelo relatório OWASP Top 10, levando em consideração que 90% das vulnerabilidades classificadas como severas são de injeção de código SQL na aplicação. Esse tipo de vulnerabilidade ainda é comumente encontrada em aplicações Web como mostrou o resultado das varreduras. As principais vulnerabilidades encontradas nas varreduras foram exploradas com autorização dos responsáveis pelos mesmos.

4.2. Exploração de injeção de SQL

Um dos sítios Web com a vulnerabilidade de injeção de SQL foi selecionado para o experimento. A vulnerabilidade foi reportada aos administradores do sítio e a exploração foi autorizada pelos mesmos. De posse da informação da URL com a falha de injeção, a ferramenta *Burp Suite* foi utilizada para interceptar a requisição HTTP destinada ao servidor Web. A requisição HTTP continha todos os parâmetros enviados ao servidor juntamente com os *cookies* da aplicação enviados naquela requisição. As informações capturadas na requisição HTTP foram salvas em um arquivo texto que posteriormente foi utilizado como fonte de informação para o arcabouço de exploração de injeção de SQL, o *SQLMap* [G. and Stampar 2013].

Utilizando o *SQLMap* juntamente as informações da requisição HTTP foi identificado o parâmetro vulnerável. O parâmetro era de controle interno da aplicação. Com a ajuda do arcabouço, combinando as técnicas de Codificação Alternativa [1] e União de Consultas [2] para injeção do código SQL, foi possível então listar 177 bases de dados hospedadas no mesmo *host* em que a aplicação alvo. Logo após foram extraídos dados sensíveis da base de dados, como usuário e o *hash* da senha administrativa. De posse do *hash* da senha foi executado um ataque de força bruta utilizando a ferramenta *John The Ripper* no modo Incremental [3] que depois de 1 dia 7 horas e 56 segundos revelou a senha correspondente ao *hash* testado. A senha descoberta foi então validada efetuando a autenticação na área administrativa do sítio Web. Verifica-se que, apenas um parâmetro não validado em uma aplicação Web, expõe ao risco várias outras aplicações e, consequentemente, toda uma infraestrutura de servidor de banco de dados incorretamente configurado.

4.3. Exploração de Quebra de Gerenciamento de Sessão

Para a exploração da vulnerabilidade de quebra de gerenciamento de sessão foi executado o ataque de roubo de sessão (*Hijacking Attack*). O ataque foi realizado utilizando a abordagem homem do meio (*man in the middle*) em uma rede local sem fio, ou seja, forçando todo tráfego entre o cliente e o *gateway* da rede a passar pela máquina do atacante. O alvo do ataque eram usuários de uma rede social conhecida. Já filtrando o tráfego da rede, utilizou-se a ferramenta *Wireshark* para visualizar os pacotes de rede e efetuar filtros devido ao grande volume de pacotes trafegando na rede. Um filtro específico então foi criado para exibir somente pacotes de rede que continham um texto específico característico de *cookies* da rede social.

Após este processo, depois de poucos minutos, foram exibidos alguns pacotes que continham os *cookies* alvo. Utilizando uma opção da ferramenta foi possível exportar o *cookie* em texto puro que posteriormente foi injetado em um navegador. Feito isso, acessando o endereço da rede social e atualizando a página após a injeção do *cookie*, obteve-se acesso completo a conta da vítima. Isso ressalta os riscos da interceptação do tráfego em redes sem fio abertas.

[1] Nesta técnica de injeção de SQL, os invasores modificam uma consulta injetando codificação alternativa, como hexadecimal, ASCII e Unicode, deste modo, podem evadir filtros criados na aplicação.

[2] União de consultas (Union Query) - Esse ataque usa o operador UNION que realiza uniões entre duas ou mais consultas.

[3] O *John The Ripper* em seu modo Incremental testa todas as combinações possíveis de caracteres para tentar quebrar a senha cifrada.

4.4. Exploração de Configuração Incorreta de Segurança

A configuração incorreta de segurança pode comprometer toda uma infraestrutura. Os ataques realizados na sessão 4.2 utilizando a injeção de SQL mostrou a possibilidade de listar e explorar 177 base de dados, podendo assim minerar dados sensíveis de todas as bases. Utilizando o arcabouço *SQLMap* foi possível constatar o usuário configurado para cada base. Todas as bases estavam configuradas para acesso apenas de um usuário do sistema gerenciador de banco de dados. Portanto, possuindo acesso a base de dados de apenas uma aplicação, o acesso para exploração e mineração das outras bases está garantido. Tal configuração comprometeu a segurança de 177 aplicações que por sua vez poderiam ser manipuladas e ter sua integridade comprometida.

4.5. Exploração de Exposição de Dados Sensíveis

Durante as varreduras em um sítio educacional, que por sua vez utilizava o CMS *Moodle*, utilizado para gerenciamento de disciplinas, foi constatado que o mesmo não utilizava o uso do acesso seguro por meio do HTTPS. Portanto, todos os dados que trafegavam entre o cliente e o servidor não possuíam nenhum tipo de encriptação. Executando o ataque homem do meio (*man in the middle*) em uma rede local onde os usuários do sítio estavam constantemente efetuando conexões com o mesmo, com a ajuda da ferramenta *wireshark* foi possível visualizar em texto plano as informações de usuário e senha de todos os usuários que efetuavam a autenticação no sítio Web. Revelando assim um grande problema em não se criptografar dados sensíveis que transitam em redes de computadores.

4.6. Exploração de Componentes com Vulnerabilidades Conhecidas

Nas varreduras executadas nos sítios Web foi encontrado um caso de uma aplicação que apresentava uma vulnerabilidade decorrente de componentes de software de terceiros. Neste caso específico, um tema para o *Wordpress*. O sítio Web analisado apresentava uma instância do CMS *Wordpress* que em seu módulo principal, na versão atualizada, não apresentava a vulnerabilidade. A falha encontrada no sítio Web foi do tipo de injeção de SQL, causada pela instalação de uma extensão que altera o tema padrão do mesmo deixando a aplicação vulnerável a ataques de injeção. E, como visto na seção 4.2, pode facilmente ser explorada. Os testes de Exploração de Componentes com Vulnerabilidades Conhecidas não foram realizados por se tratar de uma aplicação Web real e por não ter sido obtida autorização para exploração da mesma. Neste caso, as vulnerabilidades foram enumeradas e informadas aos administradores do sítio Web.

5. Discussões e Limitações

Vários problemas foram encontrados durante execução das varreduras nos sítios Web. A utilização de *Web Scanners* durante as explorações gerou um excesso de requisições HTTP para os endereços analisados causando por mais de uma vez o bloqueio do acesso a Internet por parte do provedor de acesso. Problemas como envio de *emails* em massa para usuários de aplicações Web analisadas impediram o término da explorações em sítios específicos, causando assim transtornos para os usuários das aplicações. Esses problemas diminuíram drasticamente o número de sítios Web analisados de 100 planejados para 16.

Todos os sítios analisados apresentaram vulnerabilidades, o que ressalta a falta de preocupação com segurança, principalmente de empresas de desenvolvimento de pequeno porte. Em alguns casos, mesmo após a comunicação da falha de segurança

aos responsáveis pelo desenvolvimento das aplicações, não foram realizadas correções. Acredita-se que fatores como o custo de correções de aplicações e falta de conhecimento técnico sobre as vulnerabilidades são os principais motivos da não correção das vulnerabilidades relatadas a empresas de desenvolvimento de software de pequeno porte.

Os testes de exploração de vulnerabilidades executados em cenários reais obtiveram sucesso e, em mais de um caso. A vulnerabilidade explorada poderia comprometer seriamente a aplicação e o negócio vinculada a mesma. As vulnerabilidades encontradas exemplificam quanto o desenvolvedor de aplicações Web deve estar atento as possíveis falhas durante o ciclo de desenvolvimento e mesmo de implantação do software.

6. Conclusões

A análise das vulnerabilidades Web por meio da exploração de falhas revelam os principais pontos críticos das aplicações Web. Apesar de esforços, como os da OWASP, na conscientização sobre falhas de segurança, muitas empresas de desenvolvimento ainda não aplicam os principais conceitos de segurança no desenvolvimento de suas aplicações. Fatores como esse contribuem para um crescente número de falhas básicas e simples de serem exploradas, conforme foi apresentado neste trabalho. Há a urgência de conscientizar e treinar os desenvolvedores de software sobre a importância de programação segura com o intuito de diminuir as vulnerabilidades nas aplicações Web. Como trabalhos futuros pretende-se realizar investigações de segurança de aplicações diretamente nas empresas de desenvolvimento de pequeno porte, visando identificar em quais ciclos de desenvolvimento são introduzidos as falhas e propor melhorias para aumentar a segurança das aplicações.

7. Referências

Referências

Anderson, J. (2014). John the ripper mpi patch. Acesso em: 16 fev. 2014.

Bishop, M. and Frincke, D. (2005). Teaching secure programming. *Security Privacy, IEEE*, 3(5):54–56.

CERT.BR and CGI.BR (2012). Cartilha de Segurança para Internet. Glossário.

Dacosta, I., Chakradeo, S., Ahamad, M., and Traynor, P. (2012). One-time cookies: Preventing session hijacking attacks with stateless authentication tokens. *ACM Trans. Internet Technol.*, 12(1):1:1–1:24.

G., B. D. A. and Stampar, M. (2013). Sqlmap automatic sql injection and database takeover tool @ONLINE. Acesso em: 23 ago. 2013.

Halfond, W., Viegas, J., and Orso, A. (2006). A classification of sql-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering, Arlington, VA, USA*, pages 13–15.

Hinrichs, T. L., Rossetti, D., Petronella, G., Venkatakrishnan, V. N., Sistla, A. P., and Zuck, L. D. (2013). Weblog: a declarative language for secure web development. In *Proceedings of the Eighth ACM SIGPLAN workshop on Programming languages and analysis for security*, PLAS '13, pages 59–70, New York, NY, USA. ACM.

Huluka, D. and Popov, O. (2012). Root cause analysis of session management and broken authentication vulnerabilities. In *Internet Security (WorldCIS), 2012 World Congress on*, pages 82–86.

Kolšek, M. (2002). Session fixation vulnerability in web-based applications. *Acros Security*, page 7.

Koshutanski, H. and Massacci, F. (2008). Interactive access control for autonomic systems: From theory to implementation. *ACM Trans. Auton. Adapt. Syst.*, 3(3):9:1–9:31.

Lin, X., Zavarsky, P., Ruhl, R., and Lindskog, D. (2009). Threat modeling for csrf attacks. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 3, pages 486–491.

Mello, E. R., Wangham, M. S., da Silva Fraga, J., and Camargo, E. (2006). *Segurança em Serviços Web*. Departamento de Automação e Sistemas.

Orebaugh, A., Ramirez, G., Burke, J., and Pesce, L. (2006). *Wireshark & Ethereal Network Protocol Analyzer Toolkit (Jay Beale's Open Source Security)*. Syngress Publishing.

Pelizzi, R. and Sekar, R. (2012). Protection, usability and improvements in reflected xss filters. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 5–5, New York, NY, USA. ACM.

Portswigger (2014). Burp suite. Acesso em: 16 fev. 2014.

Shirey, R. (2000). Internet Security Glossary. RFC 2828 (Informational). Obsoleted by RFC 4949.

Squyres, J. M. and Lumsdaine, A. (2004). The component architecture of open MPI: Enabling third-party collective algorithms. In Getov, V. and Kielmann, T., editors, *Proceedings, 18th ACM International Conference on Supercomputing, Workshop on Component Models and Systems for Grid Applications*, pages 167–185, St. Malo, France. Springer.

SUBGRAPH (2013). Open source scanner and testing platform to test the security of web applications @ONLINE. Acesso em: 23 ago. 2013.

Ten, T. (2013). The 2013 owasp top 10. In *AppSec USA 2013*. Owasp.

Virtue, T. (2008). *Payment Card Industry Data Security Standard Handbook*. Wiley.

Vogt, P., Nentwich, F., Jovanovic, N., Kirda, E., Kruegel, C., and Vigna, G. (2007). Cross site scripting prevention with dynamic data tainting and static analysis. In *NDSS*.

W3af (2013). Open source web application security scanner and web application attack and audit framework @ONLINE. Acesso em: 22 ago. 2013.

Wu, Y., Gandhi, R. A., and Siy, H. (2010). Using semantic templates to study vulnerabilities recorded in large software repositories. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems*, SESS '10, pages 22–28, New York, NY, USA. ACM.

Sistema de Gerenciamento de Identidades para a Rede Catarinense de Informações Municipais baseado no SAML

Emerson Souto[1,2], **Marlon Cordeiro Domenech**[1*], **Michelle Silva Wangham**[1]

[1]Laboratório de Sistemas Embarcados e Distribuídos
Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

[2]Federação Catarinense de Municípios (FECAM) – Florianópolis, SC – Brasil

souto@fecam.org.br, {marloncdomenech, wangham}@univali.br

Abstract. *The Rede Catarinense de Informações Municipais (RedeCIM) integrates many systems for supporting municipal public management. Such systems use different authentication and authorization mechanisms, what leads users to deal with different credentials for each system. This work describes a centralized identity management system based on SAML 2.0 standard, which provides Single Sign-on adequate to the requirements of RedeCIM. The satisfaction research conducted shows the approval of users and managers of RedeCIM and the software tests made show the feasibility of the solution, guaranteeing identity management on RedeCIM.*

Resumo. *A Rede Catarinense de Informações Municipais (RedeCIM) integra diversos sistemas de apoio à gestão pública municipal. Tais sistemas utilizam diferentes tecnologias e políticas de autenticação e de autorização, fazendo com que os usuários precisem lidar com diferentes credenciais para cada sistema. Este trabalho descreve um sistema de gestão de identidades centralizado baseado no padrão SAML 2.0, que provê autenticação única alinhada aos requisitos da RedeCIM. A pesquisa de satisfação realizada atestou a aprovação dos usuários e gestores da RedeCIM e os testes de software evidenciam a viabilidade da solução, garantindo a gestão de identidades na RedeCIM.*

1. Introdução

A Internet é considerada um importante meio para melhorar a eficácia e a qualidade dos serviços prestados aos cidadãos de um país [Ferreira e Araujo 2000]. O desenvolvimento de programas de Governo Eletrônico tem como princípio a utilização das modernas tecnologias de informação e comunicação (TICs) para democratizar o acesso à informação, ampliar discussões e dinamizar a prestação de serviços públicos com foco na eficiência e efetividade das funções governamentais [Dawes e Pardo 2002].

A pesquisa coordenada pelo Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (CETIC.br) sobre o uso das TICs no Brasil - TIC Domicílios e Usuários 2013, aponta um aumento de 4,1% no número de usuários de Internet no Brasil, de 2012 para 2013, totalizando aproximadamente 85,9 milhões de usuários. Segundo esta pesquisa, 68% dos brasileiros acima de 16 anos utilizaram serviços de governo eletrônico (e-Gov) nos 12 meses anteriores a pesquisa [CETIC.br 2014]. Já a pesquisa

*O autor é financiado pela CAPES.

TIC Empresas 2013, também coordenada pelo CETIC.br, indicou que 90% das empresas com acesso à Internet utilizaram os serviços de e-Gov para buscar informações ou interagir com instituições governamentais no mesmo período [CETIC.br 2013].

Após as mudanças na estrutura governamental do Brasil com a Constituição Federal de 1988, os municípios conquistaram autonomia e herdaram atribuições e responsabilidades na execução de políticas públicas em diversas áreas, tais como educação, saúde, agricultura e assistência social. Entretanto, os recursos financeiros continuaram centralizados na União e nos Estados, o que ajuda a explicar a dificuldade que os municípios, principalmente os de pequeno porte, possuem para implantar um governo eletrônico eficiente e adequado à demanda crescente da sociedade [Vedana 2001].

Diante deste cenário, a Federação Catarinense de Municípios (FECAM), a partir de 2005, criou a Rede Catarinense de Informações Municipais (RedeCIM), que integra diversas soluções tecnológicas de apoio à gestão pública municipal, promovendo o governo eletrônico nos municípios catarinenses. Estas soluções são compostas por sistemas que proveem serviços aos municípios catarinenses, por meio de uma rede colaborativa interinstitucional. Em 2013, o número de usuários com acesso aos sistemas da RedeCIM era de aproximadamente dois mil, provenientes dos 295 municípios [FECAM 2014].

Os sistemas da RedeCIM possuem diferentes tecnologias e políticas de autenticação e de autorização e seguem o modelo de gestão de identidades (*Identity Management* - IdM) tradicional (ou isolado). Os usuários normalmente precisam acessar diversos sistemas e a cada acesso é preciso realizar uma nova autenticação, o que faz com que o usuário precise lidar com um grande número de credenciais. Para a organização, há uma elevada carga administrativa em função do uso de diferentes provedores de identidade (*Identity Provider* - IdP) integrados aos diferentes sistemas. Esse fato é agravado pela alta rotatividade dos servidores públicos municipais, o que implica na necessidade constante de revogação e cadastro de identidades nos vários IdPs. Uma alternativa para minimizar este problema é a autenticação única (*Single Sign-On* - SSO), a qual permite que um usuário autenticado em um provedor de serviço (*Service Provider* - SP) seja considerado autenticado por outros SPs, normalmente utilizando um IdP para a atribuição de identificadores, emissão de credenciais e para a autenticação do usuário [Jøsang e Pope 2005].

O objetivo deste trabalho é descrever o desenvolvimento de um sistema de IdM com suporte ao SSO para a RedeCIM, visando aprimorar o processo atual de IdM. O sistema de IdM desenvolvido utiliza a especificação SAML 2.0 [OASIS 2008] como solução para troca de mensagens seguras entre os SPs da RedeCIM e o IdP. Essa opção fundamentou-se nas recomendações da arquitetura e-PING (Padrões de Interoperabilidade de Governo Eletrônico), a qual define um conjunto mínimo de premissas, políticas e especificações técnicas que regulamentam a utilização de TICs no governo federal [BRASIL 2014]. Como a FECAM é a provedora de todos os sistemas envolvidos na RedeCIM, o modelo de gestão de identidades centralizado foi o adotado. Entretanto, a escolha do SAML permite que, no futuro, o modelo de identidades federadas possa ser empregado. A solução desenvolvida foi avaliada por meio de uma pesquisa de satisfação com os usuários dos sistemas da RedeCIM e profissionais que gerenciam tais sistemas, a qual demonstrou a satisfação dos usuários e administradores com o uso do sistema de IdM. Também foram realizados testes de integração, usabilidade e de segurança, os quais indicaram o atendimento aos requisitos funcionais e não funcionais da solução.

Este trabalho assemelha-se à Comunidade Acadêmica Federada (CAFe)[1], a qual utiliza o modelo de gestão de identidades federadas no contexto das instituições de ensino e pesquisa brasileiras. A CAFe provê a autenticação única na federação e faz uso do padrão SAML 2.0, conforme implementado no framework Shibboleth. Este trabalho difere da CAFe ao adotar o modelo de IdM centralizado e utilizar a implementação do framework SimpleSAMLphp[2] da especificação SAML 2.0.

O restante do artigo está organizado da seguinte maneira. A Seção 2 apresenta a fundamentação teórica sobre IdM e uma comparação entre as soluções tecnológicas que podem ser utilizadas para prover o SSO na RedeCIM. A Seção 3 apresenta a solução proposta, seguida pela Seção 4, que apresenta os experimentos e a avaliação da solução. A Seção 5 apresenta as conclusões e as perspectivas de trabalhos futuros.

2. Gestão de Identidades Digitais

A gestão de identidades (*Identity Management* – IdM), pode ser entendida como o conjunto de processos e tecnologias usados para garantir a identidade de uma entidade ou de um objeto, garantir a qualidade das informações de uma identidade (identificadores, credenciais e atributos) e para prover procedimentos de autenticação, autorização, contabilização e auditoria [ITU 2009]. As entidades envolvidas em um sistema de IdM são o provedor de identidades (IdP), responsável por gerar identidades, manter a base de dados de usuários do domínio e validar suas credenciais; o provedor de serviços (SP), que oferece recursos ou serviços aos usuários; e o usuário, que utiliza um serviço fornecido por um SP [Bhargav-Spantzel et al. 2007].

Os sistemas de IdM podem ser caracterizados por quatro modelos distintos: tradicional, centralizado, federado e centrado no usuário. No modelo tradicional, o SP atua como SP e IdP, sendo que o usuário possui um identificador e credenciais únicas para cada SP que acessa e não há compartilhamento de identidades entre organizações. O modelo centralizado utiliza apenas um IdP, no qual usuários e SPs devem confiar. Existe o compartilhamento de identidades dos usuários entre SPs e é possível implementar o SSO. O modelo federado promove a descentralização do trabalho do IdP por vários IdPs diferentes, localizados em domínios administrativos de segurança diferentes. Identidades emitidas e verificadas por um IdP são aceitas em outro domínio administrativo com base em acordos de confiança estabelecidos entre IdPs e SPs, formando assim uma federação de identidades. O modelo federado também permite o SSO. O modelo centrado no usuário tem por objetivo dar maior controle ao usuário sobre as transações envolvendo os seus dados de identidade, não deixando tais dados sob controle total dos IdPs e SPs [Wangham et al. 2010, Bhargav-Spantzel et al. 2007].

O modelo de IdM que é utilizado nos SPs da RedeCIM é o tradicional, tendo o usuário que lidar com diferentes identidades em diferentes sistemas. Para o sistema de IdM descrito neste trabalho, optou-se pelo modelo de IdM centralizado, pois a FECAM, fornecedora e mantenedora dos sistemas utilizados nos municípios, precisa manter o controle do processo de autenticação e das identidades digitais ligadas às diversas instituições.

Para definição da solução tecnológica que possibilita a autenticação SSO e que esteja alinhada aos requisitos da RedeCIM, foi conduzido um estudo envolvendo o SAML[3],

[1] Disponível em: http://www.rnp.br/servicos/servicos-avancados/cafe
[2] https://simplesamlphp.org/
[3] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

OpenID[4], OAuth[5], Kerberos[6], CAS (Central Authentication Service)[7] e OpenLDAP[8]. Na Tabela 1, é possível analisar as soluções diante das seguintes características: modelos de IdM; mecanismos de autenticação suportados; a capacidade da ferramenta se integrar com outras tecnologias e aplicações e, por fim, a dificuldade de uso da ferramenta.

Tabela 1. Comparação das ferramentas estudadas

Ferramenta	Modelo de IdM	Mecanismo de Autenticação	Integração com outras tecnologias	Uso
OpenID	Centralizado, Federado	Escolha do Provedor	Apache, C++, Java e PHP	Fácil
OAuth	Centralizado, Federado	Escolha do Provedor	Java, .Net, C# e PHP	Fácil
CAS	Centralizado	JDBC, LDAP, Radius, X.509, RET API	Apache, Java, PERL, PHP, ASP, Shibboleth	Fácil
OpenLDAP	Centralizado	Cryrus Sasl Kerberus V, GSSAPI, Digst-MD5	FTP, Apache, SQUID e SAMBA	Moderado
Kerberos	Centralizado, Federado	Kerberos	OpenLDAP	Moderado
SAML	Centralizado, Federado	Escolha do Provedor	OpenID, Oauth, CAS, JOSSO e OpenAM	Fácil

O padrão SAML 2.0, por ser uma solução que possibilita implementar o modelo de IdM centralizado e por possuir um suporte ao SSO, se mostrou a solução mais adequada. Uma das vantagens desta especificação é a disponibilidade do framework SimpleSAMLphp para a implementação do sistema de IdM, de acordo com o SAML 2.0. Outra vantagem do uso do SAML é a possibilidade de implementar o modelo de IdM federado. Um ponto importante nesta escolha é que a arquitetura e-PING recomenda o SAML como ferramenta de autenticação e autorização de acesso XML, sendo um passo importante para prover a interoperabilidade entre diferentes aplicações de governo eletrônico. Por fim, segundo o relatório da OECD [OECD 2011], o padrão SAML é o mais adotado nas estratégias nacionais de IdM nos países avaliados.

O padrão SAML define um framework baseado em XML para a troca de informações de segurança entre parceiros de negócio, sendo independente de mecanismos e protocolos proprietários. Essas informações de segurança são expressas por meio de asserções SAML portáveis entre os sistemas participantes. Tais sistemas, que podem estar localizados em domínios de segurança distintos, podem confiar nas asserções trocadas entre si, sendo possível a troca de informações de atributos, autenticação e autorização acerca de um sujeito. Para isso, o padrão SAML define uma sintaxe precisa e regras para requisição, criação, comunicação e uso das asserções SAML [OASIS 2008].

A arquitetura do SAML possui componentes que permitem atender a diversos casos de uso diferentes como, por exemplo, o SSO para clientes web que possuem ou não um navegador web. Os conceitos básicos da arquitetura do SAML são [OASIS 2008]: (i) Asserções, que carregam sentenças sobre um *principal* que uma *Asserting Party* afirma que são verdadeiras; (ii) Protocolos, que definem como se dá a requisição e resposta de informações de segurança; (iii) Ligações, que definem como as mensagens de um

[4]http://openid.net/
[5]http://tools.ietf.org/html/rfc6749
[6]http://web.mit.edu/kerberos/
[7]http://www.jasig.org/cas
[8]http://www.openldap.org/

protocolo SAML serão transportadas usando um determinado protocolo de comunicação; e (iv) Perfis, que definem restrições ao conteúdo das asserções, protocolos e ligações, com a intenção de atender a um caso de uso específico, visando a interoperabilidade.

3. Solução Proposta: Sistema de IdM Centralizado para a RedeCIM

Conforme ilustrado na Figura 1, o sistema de gerenciamento de identidades interage com três atores que representam usuários dos perfis Administrador do IdP, Administrador da Instituição e do Colaborador (servidor público municipal ou colaborador das instituições municipalistas). O grande número de colaboradores que possuem acesso a mais de um SP indica que o sistema deve possibilitar a autenticação única (SSO). Desta forma, um usuário já autenticado no IdP não necessitará de nova autenticação para acessar recursos de outros SPs que integram o círculo de confiança. Neste cenário, a FECAM atua como desenvolvedora dos sistemas oferecidos às instituições municipalistas por meio da RedeCIM e como administradora do IdP, enquanto que as instituições municipalistas administram os SPs que implementam os sistemas fornecidos pela FECAM.

O ator Administrador do IdP é o responsável na FECAM por administrar o IdP no qual os usuários irão se autenticar, sendo parte de suas atribuições o cadastro de SPs para estabelecimento do círculo de confiança e o gerenciamento dos usuários que se autenticam no IdP. O representante legal da instituição municipalista (prefeito ou presidente de câmara de vereadores) é responsável por indicar, por meio de um formulário assinado, o Administrador da Instituição, o qual será responsável por cadastrar e gerenciar usuários Colaboradores. Estes últimos são os servidores das instituições que utilizarão os SPs. O ator Administrador do SP é responsável pela gerência de um SP, visando manter os serviços providos ativos e manter o círculo de confiança entre o IdP e o SP.

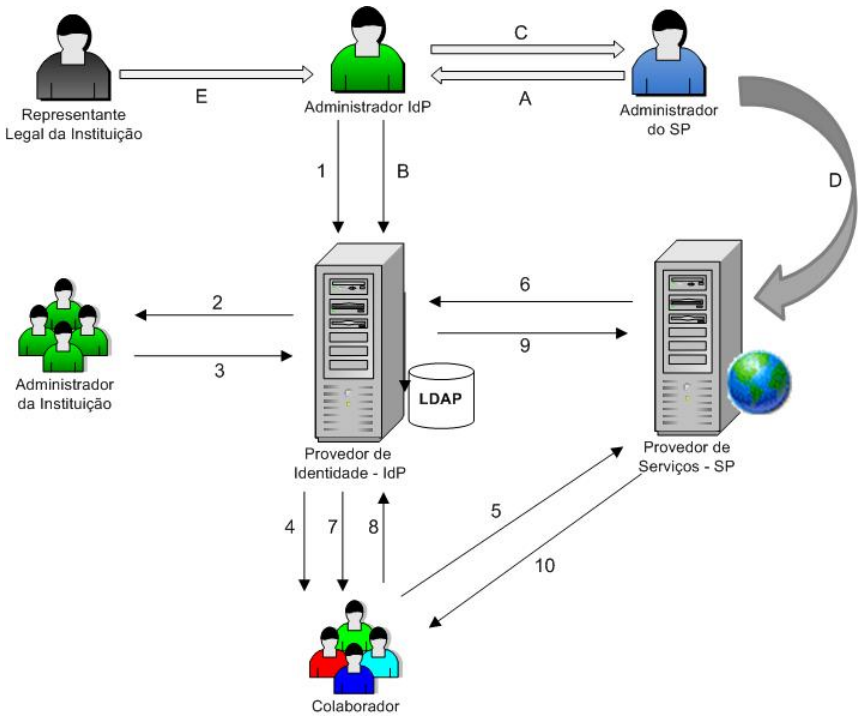


Figura 1. Visão Geral do Sistema de Gestão de Identidades para a RedeCIM

A configuração da relação de confiança entre o IdP e os SPs é realizada *a priori*

por meio da troca de informações, conforme ilustrado na Figura 1 e descrito a seguir:

A. O Administrador do SP, ator que não se relaciona diretamente com o sistema, envia os metadados do SP para o Administrador do IdP;
B. O Administrador do IdP registra os metadados do SP no IdP, visando aceitar seus pedidos de autenticação;
C. O Administrador do IdP disponibiliza os metadados do IdP para o Administrador do SP; e
D. O Administrador do SP registra no SP os metadados do IdP, fechando assim o círculo de confiança entre SP e IdP.

O passo (E) é referente à indicação do ator Administrador da Instituição. Este processo consiste no encaminhamento de um formulário assinado pelo representante legal da instituição indicando quem será o Administrador da Instituição. O formulário deve conter as informações cadastrais do Administrador da Instituição para registro no sistema. A seguir, é descrito o fluxo de mensagens entre os atores e o sistema de IdM da RedeCIM:

1. O Administrador do IdP cadastra o Administrador da Instituição no IdP;
2. O IdP encaminha para o e-mail do Administrador da Instituição as credenciais de autenticação, as quais precisarão ser modificadas no primeiro acesso;
3. O Administrador da Instituição cadastra usuários do perfil Colaborador, que são os servidores das instituições que utilizarão os serviços do SP;
4. O IdP encaminha para o e-mail do Colaborador cadastrado as credenciais de autenticação, as quais precisarão ser modificadas no primeiro acesso;
5. Após estar cadastrado, o Colaborador realiza uma tentativa de acesso ao serviço (SP) desejado, por meio de um navegador Web;
6. O SP redireciona o navegador do usuário para o IdP, para que o Colaborador se autentique;
7. O IdP solicita, por meio de uma página web, que o Colaborador proceda com a autenticação;
8. O Colaborador procede com a autenticação (utilizando CPF e senha), inserindo na página web apresentada as suas credenciais de autenticação;
9. Caso a autenticação do Colaborador seja bem sucedida, o IdP gera uma asserção SAML com os atributos do usuário autenticado e a envia ao SP; e
10. Se autorizado pelo SP, este disponibiliza ao Colaborador os recursos solicitados.

Para o funcionamento do sistema de gerenciamento de identidades, foram estabelecidas especificações técnicas que definem os atributos dos usuários e os metadados do IdP e dos SPs. A lista de atributos dos usuários é disponibilizada aos SPs pelo IdP, utilizando o formato *urn:oid:*, conforme o padrão SAML 2.0. Essa lista de atributos[9] é apresentada na Tabela 2.

Os metadados utilizados para o estabelecimento e manutenção do círculo de confiança entre IdP e SPs estão no formato SAML 2.0 Metadata [OASIS 2009], padrão estabelecido pela OASIS e também recomendado pela arquitetura e-PING.

O documento de metadados fornecido pelo IdP inclui um elemento <IDPSSO-Descriptor>, que contém informações dos elementos <KeyDescriptor>(adequado para

[9]Tomou como base a RFC 2798 (Definição da classe de objetos LDAP inetOrgPerson), disponível em: http://tools.ietf.org/html/rfc2798.

Tabela 2. Atributos do usuário

Atributo	Descrição	Fonte
Cn	Nome do usuário	inetOrgPerson
Sn	Sobrenome do usuário	inetOrgPerson
Mail	Endereço de e-mail do usuário	inetOrgPerson
Uid	Identificador único do usuário dentro da federação (Número do CPF)	inetOrgPerson
o	Nome do órgão (instituição) do usuário	inetOrgPerson

utilização de criptografia XML, sendo utilizado quando a entidade renuncia ao uso de TLS/SSL) e <SingleSingOnService>(responsável pela autenticação SSO). Estes metadados devem incluir um ou mais <NameIDFormat>, os quais indicam os formatos de valores suportados pelo elemento identificador do usuário (<NameID>).

O documento de metadados fornecido por um SP deve incluir um elemento <SPSSODescriptor>, que contém informações dos elementos <KeyDescriptor>e <AssertionConsumerService>(elemento que possibilita ao SP receber asserções do IdP). Os metadados incluem também um ou mais <NameIDFormat>e um ou mais <AttributeConsumingService>(descreve o(s) serviço(s) oferecido(s) e as suas necessidades de atributos). Os metadados devem conter um nome descritivo do serviço oferecido, o qual deve ser colocado em um elemento <ServiceName>e no respectivo <AttributeConsumingService>.

Para o desenvolvimento do IdP, foram priorizadas ferramentas e tecnologias de software livre, conforme recomendado pelo Comitê de Executivo do Governo Eletrônico Brasileiro. O sistema foi construído utilizando a estrutura de diretório LDAP (Lightweight Directory Access Protocol) para armazenamento dos dados dos usuários e serviços. No desenvolvimento do sistema, foi escolhida a linguagem PHP utilizando o Zend Framework 2, por esta ferramenta possuir uma ampla biblioteca de comunicação com o LDAP. Também foi utilizado o Framework Javascript jQuery, em função deste ser amplamente utilizado para o desenvolvimento de aplicações RIA (Rich Internet Application) na FECAM. Para prover a troca de informações de autenticação e de autorização com a especificação SAML 2.0 e para construir o ambiente de autenticação única (SSO), foi utilizada a ferramenta SimpleSAMLphp.

Como protocolo de segurança na transferência de dados foi utilizado o TLS 1.2 (Transport Layer Security). As conexões são criptografadas utilizando o AES_256_CBC, com SHA1 para mensagem de autenticação e DHE_RSA como mecanismo de troca de chaves. O certificado é emitido por StarCom[10], por ser um certificado de classe 1 gratuita e aceito pelos principais navegadores do mercado. Contudo, para utilização do IdP na FECAM será necessário adquirir um certificado confiável, emitido pela AC da ICP Brasil. A hospedagem dos sistemas foi feita em dois servidores com sistema operacional Linux, com servidor web Apache 2 e PHP 5.

4. Experimentos e Avaliação

Para avaliar a aplicabilidade do sistema de IdM, foi desenvolvido o IdP e um protótipo de SP. Foram executados 14 testes de software para avaliar as funcionalidades do sistema de IdM, conforme definidas nos casos de uso, nos requisitos funcionais e não funcionais e nas regras de negócio. Foram realizados testes nos níveis de sistema, integração, portabilidade e segurança, os quais permitiram verificar o atendimento aos requisitos elicitados.

[10]Certificado disponível em: http://www.startssl.com/?app=1

O segundo experimento consistiu na aplicação de uma pesquisa de satisfação, realizada entre 14 e 24 de outubro de 2013. Experimentos foram realizados por técnicos da FECAM, do CIGA (Consórcio de Informática na Gestão Pública Municipal) e das associações de municípios que gerenciam os usuários dos sistemas da RedeCIM. Foram elaborados três roteiros para que os profissionais soubessem os objetivos do IdP e as funcionalidades que estes deveriam testar e avaliar no sistema, porém, nenhuma informação detalhada de como o sistema foi desenvolvido foi passada. A diferença entre cada um dos roteiros elaborados são as questões, as quais são direcionadas conforme o perfil de acesso dos avaliadores do sistema desenvolvido (Administrador do IdP, Administrador da Instituição e Colaborador). Após seguir o roteiro de testes, os avaliadores responderam a uma pesquisa de satisfação que, além de avaliar a satisfação dos usuários, procurou avaliar o perfil e o nível de conhecimento destes acerca de alguns conceitos de IdM.

O experimento foi executado por 5 profissionais de TI que trabalham na FECAM e no CIGA que atuam no gerenciamento de serviços e de usuários dos sistemas disponibilizados, por 18 profissionais que atuam como gerente de contas de usuários dos serviços e mais 18 profissionais que atuam como usuários consumidores dos serviços disponibilizados nas instituições, totalizando 41 avaliadores. Dos avaliadores, 51% atuam na área de TI, 27% na área administrativa e os outros 22% atuam em áreas diversas, tais como economia, assistência social e comunicação. Dentre os avaliadores, 34% são usuários de mais de 5 sistemas e outros 49% utilizam entre 1 e 5 sistemas, números que demonstram a necessidade de ferramentas como o IdP para auxiliar na IdM das instituições.

A pesquisa foi aplicada em um grupo experiente de usuários dos serviços da RedeCIM. Do total de avaliadores, 39% atuam a mais de 5 anos nesta função, 34% atuam entre 1 e 5 anos e 27% atuam a menos de 1 ano. Se considerado apenas os avaliadores que atuam como administrador de usuários, que totalizam 23 avaliadores, o grau de experiência aumenta, pois 47,82% atuam neste papel a mais de 5 anos e apenas 17,39% atuam neste papel a menos de 1 ano.

Os avaliadores foram questionados a respeito do seu conhecimento sobre o conceito de autenticação única (SSO). Somente 41% dos avaliadores conhecem este conceito. Já o percentual de conhecimento sobre autenticação centralizada foi de 61% e o conhecimento sobre autenticação federada foi de 52%. A questão sobre o conceito de autenticação federada foi aplicada apenas aos 23 avaliadores do perfil de Administrador do IdP e Administrador da Instituição, portanto representam o conhecimento apenas dos avaliadores que gerenciam usuários dos serviços da RedeCIM. Dentre os avaliadores foi possível verificar o pouco conhecimento deles quanto aos conceitos na área de IdM, principalmente se considerarmos o tempo experiência da atuação dos avaliadores neste papel. Vale ressaltar que a pesquisa foi respondida na sua maioria por técnicos das instituições que vivenciam a administração de sistemas no seu dia-a-dia.

Considerando o total de avaliadores, 93% executaram os experimentos com sucesso, sendo que os 7% restantes indicaram erros de operação do sistema, resolvidos pelas mensagens de erro fornecidas. A respeito da satisfação dos usuários quanto as mensagens de erro, 49% dos usuários consideraram que as mensagens foram suficientes para apontar o problema ocorrido, outros 15% consideraram que as mensagens foram parcialmente suficientes e um usuário (2%) não considerou as mensagens suficientes para entender o problema. Os 34% dos usuários que restaram não tiveram problemas, portanto

não avaliaram esta questão.

Sobre o ponto de vista dos avaliadores quanto a apresentação das informações do protótipo, a maioria (76%) considerou as informações claras e compreensíveis, 22% consideraram as informações parcialmente claras e apenas 2% consideraram as informações não compreensíveis. Quanto a rapidez na execução dos serviços, 90% dos avaliadores consideraram que o IdP respondeu rapidamente as suas solicitações.

Em média, 83% dos avaliadores (perfis Administrador do IdP e Administrador da Instituição) opinaram que o gerenciamento de usuários utilizando o modelo centralizado com o IdP traz maior rapidez. Na média dos três perfis, 88% dos avaliadores opinaram que a utilização do IdP traz maior flexibilidade no acesso aos serviços. Entretanto, ao considerar apenas as opiniões dos avaliadores do perfil Colaborador, o percentual de satisfação neste quesito baixa para 83%, sendo que 11% não possuem opinião formada.

Sobre a opinião dos avaliadores referente à segurança da utilização do IdP para autenticação centralizada dos usuários, os resultados apontam que 71% dos avaliadores indicam que traz maior segurança, 24% dos avaliadores não tinham uma opinião formada e apenas 5% opinaram que o IdP não traz segurança.

Observou-se também que 40% dos Administradores do IdP gostariam de utilizar o IdP para a gestão dos usuários dos serviços da FECAM e do CIGA, enquanto que 60% não possuem opinião formada. Dentre os avaliadores do perfil Administrador da Instituição, 89% opinaram que gostariam de utilizar o IdP e 11% não possuem opinião formada. Dentre os avaliadores do perfil Colaborador, 78% opinaram que gostariam de utilizar o IdP para realizar a autenticação para acesso aos serviços, 17% não possuem opinião formada e apenas um avaliador (5%) opinou que não gostaria de utilizar o IdP.

O gráfico da Figura 2 apresenta os resultados da opinião dos avaliadores dos perfis Administrador do IdP e Administrador da Instituição. Nos dois perfis, a maioria dos avaliadores opinou que a utilização do IdP facilitaria o processo de gestão de usuários e traria maior satisfação aos usuários que utilizam os serviços disponibilizados pela FECAM.

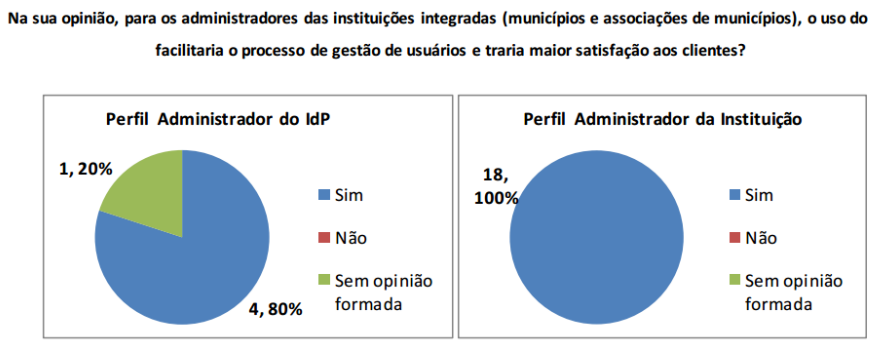


Figura 2. Opinião sobre a facilidade na gestão e na satisfação dos usuários

5. Conclusão

Este trabalho descreveu o desenvolvimento de um sistema de IdM centralizado para a RedeCIM, baseado na especificação SAML 2.0 e que segue as recomendações da arquitetura e-PING para sistemas de governo eletrônico. Com este sistema, foi provida a autenticação única (SSO) para os usuários da RedeCIM. O sistema foi avaliado por

meio de uma pesquisa de satisfação aplicada em três grupos de usuários que representam os perfis de usuários que interagem com o sistema de IdM. Os resultados da pesquisa de satisfação demonstram que os avaliadores, em sua grande maioria, ficaram satisfeitos com o uso do sistema de IdM, principalmente os administradores de serviços, que com o uso do gerenciador de identidades centralizado no IdP podem realizar a gestão dos serviços e usuários de uma forma mais simples e segura. Também foram executados casos de teste para verificação do atendimento aos requisitos funcionais e não funcionais, obtendo resultados positivos. A principal contribuição deste trabalho foi oferecer uma solução tecnológica viável para o aprimoramento do processo de IdM da FECAM, alinhada aos padrões de Governo Eletrônico do Brasil.

Como trabalho futuro pretende-se desenvolver um sistema que possibilite que os cidadãos que interagem com os portais integrados da RedeCIM consigam se autenticar nos serviços disponibilizados utilizando as credenciais de suas contas pessoais, como contas no Google, Yahoo e Facebook, desta forma facilitando o acesso sem a necessidade de novo cadastro de informações.

Referências

Bhargav-Spantzel, A., Camenisch, J., Gross, T., e Sommer, D. (2007). User centricity: a taxonomy and open issues. *Journal of Computer Security*, 15(5):493–527.

BRASIL (2014). e-ping: Padrões de interoperabilidade de governo eletrônico.

CETIC.br (2013). Tic empresas 2013.

CETIC.br (2014). Tic domicílios e usuários 2013.

Dawes, S. S. e Pardo, T. A. (2002). Building collaborative digital government systems. In McIver, WilliamJ., J. e Elmagarmid, A., editors, *Advances in Digital Government*, volume 26 of *Advances in Database Systems*, pages 259–273. Springer US.

FECAM (2014). Guia dos municípios catarinenses 2013/2014.

Ferreira, S. G. e Araujo, E. A. (2000). Modernização da gestão: E-governo: O que ensina a experiência internacional. *INFORME-SF*, (17):1–6.

ITU (2009). Ngn identity management framework. Recommendation Y.2720.

Jøsang, A. e Pope, S. (2005). User centric identity management. In *Proceedings...* AusCERT Asia Pacific Information Technology Security Conference 2005.

OASIS (2008). Security assertion markup language (saml) v2.0 - technical overview.

OASIS (2009). Metadata for the oasis security assertion markup language (saml) v2.0 - errata composite - working draft 04.

OECD (2011). National strategies and policies for digital identity management in OECD countries. *OECD Digital Economy Papers*, (177).

Vedana, C. (2001). *Federalismo: autonomia tributária formal dos municípios*. PhD thesis, Universidade Federal de Santa Catarina, Centro de Ciências Jurídicas. Programa de Pós-Graduação em Direito.

Wangham, M. S., de Mello, E. R., da Silva Böger, D., Guerios, M., e da Silva Fraga, J. (2010). Gerenciamento de identidades federadas. In *Minicurso – SBSeg 2010*.

Implementação Eficiente de Algoritmos para Teste de Primalidade

Bruno C. Dias Ribeiro[1], **Diego F. Aranha**[2]

[1]Departamento de Ciência da Computação - Universidade de Brasília (UnB)
70910-900 – Brasília – DF – Brasil

[2]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
13083-852 – Campinas – SP - Brasil

brunocapu@gmail.com, dfaranha@ic.unicamp.br

Abstract. *The development of cryptography, public key cryptography in particular, was a crucial factor for the growth and popularization of computer networks. It was responsible for demands of secure email communication, e-commerce, digital signatures and certification. The proper use of cryptographic techniques requires the development of efficient implementations capable of running in all kind of devices, which are more and more integrated into people's lives. The generation of cryptographic keys, for example, is not only a critical security operation but also has a high computational cost in the RSA algorithm. This work aims to study techniques for primality testing, the core element of the key generation process in this algorithm. Our focus is on the implementation, optimization and performance analysis of Simplified Quadratic Frobenius Test, a primality test from 2005 which did not receive enough attention in the literature. The achieved results points to a 30% efficiency increase for integers up to 256* bits*, a positive outcome regarding the feasibility of reducing the computational cost of generating prime numbers.*

Resumo. *O desenvolvimento da criptografia, em especial a criptografia de chave assimétrica, foi fator determinante para o crescimento e popularização das redes de computadores. Foi responsável pela viabilização de demandas como comércio e correio eletrônicos, assinaturas e certificações digitais. O uso adequado de técnicas criptográficas requer o desenvolvimento de implementações eficientes que sejam capazes de serem executadas em diversos tipos de dispositivos, que cada vez mais se incorporam à vida das pessoas. A geração de chaves criptográficas, por exemplo, é uma operação não só crítica quanto à segurança, mas também de alto custo computacional no algoritmo RSA. Este trabalho tem como objetivo estudar técnicas para teste de primalidade, elemento que compõe o núcleo do processo de geração de chaves nesse algoritmo. É dado enfoque na implementação, otimização e análise de desempenho do Teste de Frobenius Quadrático Simplificado, um teste de primalidade de 2005 e pouco explorado. As análises apontam uma eficiência 30% maior para inteiros de até 256* bits*, resultados positivos quanto à viabilidade da redução do custo computacional da geração de números primos.*

1. Introdução

Após as recentes denúncias sobre os casos de vigilantismo global por parte do Governo norte-americano, a criptografia e segurança digital têm sido um tema constante na mídia mundial. No momento, há grande discussão a respeito da liberdade e privacidade individual e a criptografia é elemento fundamental para a garantia de tais direitos civis no cenário de telecomunicações globalizado.

O tamanho das chaves de sistemas criptográficos de chave pública como o RSA tendem a crescer consideravelmente nos próximos 10 anos. Na maioria dos casos há exigência dessas chaves serem geradas em sigilo, então tais aplicações devem ser executadas também em ambientes seguros como *smartcards* ou dispositivos blindados. Esses dispositivos normalmente possuem poder computacional muito limitado, tornando a geração de chaves de alto nível de segurança uma operação demorada e até mesmo com tempo de execução proibitivo.

Este trabalho crê na premissa de que a difusão de criptografia depende de soluções práticas e eficientes. Para isso, o seu objetivo geral é reduzir o custo computacional na geração de chaves criptográficas. Como objetivos específicos, tem-se a análise, implementação e otimização de algoritmos de teste de primalidade, que são responsáveis por maior porcentagem do tempo de geração de chaves no algoritmo RSA.

2. Teste de Primalidade

A fatoração de números grandes é tida como um problema intratável, porém percebeu-se que saber apenas quanto à primalidade de um número é uma matéria distinta. Teste de primalidade pode ser modelado como um problema de decisão em que tem resultado SIM para primo e NÃO para composto. Os testes são divididos em duas categorias:

- Testes Determinísticos: são testes que asseguram certeza matemática para todas as respostas.
- Testes Probabilísticos: são testes que contém probabilidade de erro, de incerteza. Escolhe-se um candidato aleatório e aplica-se a ele critérios que possam refutar ou confirmar certa propriedade. A cada rodada do teste, o candidato adquire maior probabilidade para uma resposta correta.

Apesar de existir teste determinístico com tempo de execução polinomial como o AKS (Agrawal et al. 2004), os testes probabilísticos, com segura margem de erro delimitada em 2^{-80} ou 2^{-100} (ISO/IEC 18032 2005), ainda são mais eficientes. Algoritmos probabilísticos, por sua vez, podem ser classificados em dois tipos:

- Algoritmos de Monte Carlo: podem produzir resultado incorreto.
- Algoritmos Las Vegas: podem causar falha ao tentar produzir um resultado.

Para problemas de decisão, algoritmos de Monte Carlo são usualmente classificados como:

- Monte Carlo com viés positivo: é sempre correto quando retorna SIM.
- Monte Carlo com viés negativo: é sempre correto quanto retorna NÃO.

2.1. Teste de Miller-Rabin

O teste de Miller-Rabin (Miller 1976; Rabin 1980) é um algoritmo de Monte Carlo com viés positivo para compositividade, em outras palavras, quando retorna COMPOSTO, o número é provado composto, quando retorna PRIMO, o número é provavelmente primo. É o teste mais utilizado nas soluções de criptografia e um dos mais eficientes da atualidade (Dietzfelbinger 2004), portanto será usado como comparativo em seções posteriores. A rodada do teste de Miller-Rabin é apresentada no Algoritmo 1.

Algoritmo 1 Teste de primalidade de Miller-Rabin

Input: Inteiro ímpar $n \geq 3$.
Output: Ou PRIMO ou COMPOSTO.

1: Escreve $n-1$ na forma $2^k m$ {com k máximo e m ímpar}
2: $a \leftarrow$ inteiro aleatório tal que $1 \leq a \leq n-1$
3: $b \leftarrow a^m \bmod n$
4: **if** $b = 1$ **then return** PRIMO
5: **for** $i \leftarrow 1$ **to** $k-1$ **do**
6: **if** $b = -1$ **then return** PRIMO
7: **else** $b \leftarrow b^2 \bmod n$
8: **end for**
9: **return** COMPOSTO

2.2. Teste de Frobenius Quadrático Simplificado

O Teste de Frobenius Quadrático Simplificado (TFQS) (Seysen 2005) é baseado no Teste de Frobenius Quadrático (TFQ) (Grantham 1998). Existem outros testes correlatos também com base no trabalho de Grantham, como o Teste de Siguna Müller (Müller 2001; Müller 2003), que possui baixa probabilidade de erro, mas comportamento assintótico e custo computacional semelhante ao TFQ, e o teste TFQEpc (Damgard and Frandsen 2003), que é otimizado para o pior caso e tem baixo custo, no entanto requer um pré-cálculo de resíduos cúbicos com custo em torno de 2 testes de Miller-Rabin. Dentre esses, o TFQS é o que apresenta melhor desempenho teórico, portanto foi o teste proposto para implementação e análise. Seu tempo de execução foi reduzido para duas rodadas de Miller-Rabin e probabilidade de erro, para o pior caso, de 2^{-12t}, sendo t o número de rodadas.

O TFQS é um algoritmo de Monte Carlo com viés positivo para compositividade, usa polinômios quadráticos e o automorfismo de Frobenius. Seja $q = p^m$ a potência de um primo p e o corpo finito $F_q = G(q)$, G a extensão de Galois. O **automorfismo de Frobenius** ϕ_q é o mapeamento na bijeção

$$\begin{aligned} \phi : F &\to F \\ z &\mapsto z^q \end{aligned}$$

para todo $z \in F$. Para um n natural ímpar e c uma unidade módulo n, $R(n,c)$ denota o anel polinomial $\mathbb{Z}_n[x]/(f(x) = x^2 - c)$, e $R(n,c)^*$ o grupo multiplicativo em $R(n,c)$. Sendo $f(x)$ um polinômio mônico quadrático em $\mathbb{Z}_n$, se n é primo e $f(x)$ é irredutível em $\mathbb{Z}_n$, então este anel é equivalente ao corpo finito $G(n^2)$. Felizmente, esses polinômios são facilmente encontrados: para n primo, um polinômio quadrático em $\mathbb{Z}_n$ é irredutível se e

somente se seu discriminante, Δ, é um resíduo não-quadrático módulo n, ou seja, para o polinômio mônico $x^2 - bx - c$, $\left(\frac{b^2+4c}{n}\right) = -1$.

$G(n^2)$ é cíclico de ordem $n^2 - 1$, então qualquer $z \in R(n,c)^*$ deve ter uma ordem dividindo $n^2 - 1$. Também possui um automorfismo natural, chamado "conjugado" em $R(n,c)$, que deve ser equivalente ao automorfismo de Frobenius $z \to z^n$ em $G(n^2)$ para n primo. Todas q-ésimas raízes de unidade z, se existir, satisfazem $\Phi_q(z) = 0$ para o q-ésimo polinômio ciclotômico Φ_q.

Para um polinômio $z = ax + b$, define-se o seguinte homomorfismo multiplicativo em $R(n,c)$:

$$\begin{array}{lcll} \bar{\cdot} & : & R(n,c) \to R(n,c), & \bar{z} = b - ax \quad \text{(conjugado);} \\ N(\cdot) & : & R(n,c) \to \mathbb{Z}_n, & N(z) = \bar{z} \cdot z = b^2 - ca^2 \quad \text{(norma).} \end{array}$$

A notação (a/n) se refere ao símbolo de Jacobi. O Algoritmo 2 (MR2) é um crivo inicial equivalente a uma rodada do teste de Miller-Rabin com uma base pequena. Retorna ou uma raiz oitava primitiva de unidade em uma extensão quadrática conveniente de $\mathbb{Z}_n$ ou uma prova de que n é composto.

Algoritmo 2 Teste de Miller-Rabin com base dois ou não-resíduo pequeno

Input: Inteiro ímpar n.
Output: Ou COMPOSTO ou um inteiro c tal que $(c/n) = -1$ e $\epsilon \in R(n,c)$ com $\epsilon^4 = -1$ (i.e. $\Phi_8(\epsilon) = 0$).

1: **if** $n = 3 \pmod 4$ **then**
2: $\alpha \leftarrow 2^{\frac{n-3}{4}} \pmod n$
3: **if** $2\alpha^2 \neq \pm 1 \pmod n$ **then return** COMPOSTO
4: **else return** $c \leftarrow -1$, $\epsilon \leftarrow \alpha + \alpha x$
5: **end if**
6: **if** $n = 5 \pmod 8$ **then**
7: $\alpha \leftarrow 2^{\frac{n-1}{4}} \pmod n$
8: **if** $\alpha^2 \neq -1 \pmod n$ **then return** COMPOSTO
9: **else return** $c \leftarrow 2$, $\epsilon \leftarrow \frac{1+\alpha}{2} x$
10: **end if**
11: **if** $n = 1 \pmod 8$ **then**
12: **if** n é um quadrado perfeito **then**
13: **return** COMPOSTO
14: **else**
15: $c \leftarrow$ valor aleatório pequeno tal que $(c/n) = -1$
16: $\alpha \leftarrow c^{\frac{n-1}{8}} \pmod n$
17: **if** $\alpha^4 \neq -1 \pmod n$ **then return** COMPOSTO
18: **else return** c, $\epsilon \leftarrow \alpha$
19: **end if**
20: **end if**

Na verdade, o Algoritmo MR2 realiza uma rodada de Miller-Rabin com base 2 caso $n \neq 1 \pmod 8$ e com base c caso contrário. A relação $\epsilon^4 = -1 \pmod{R(n,c)}$ é

evidente caso $n = 1 \pmod 8$ e facilmente verificável pelas representações padrões $(\pm 1 \pm \sqrt{-1})/\sqrt{2}$ das quatro raízes oitavas primitivas de unidade nos outros casos (Seysen 2005).

Como foi citado, o conjugado também é um automorfismo em $R(n, c)$. Suponha o grupo G. Dois elementos a e b são ditos conjugados se existe um elemento $g \in G$ tal que $gag^{-1} = b$. Isso mostra que o conjugado é uma relação de equivalência e toda conjugação é um automorfismo. Logo, a não verificação do automorfismo de Frobenius $(z^n \neq \bar{z})$ é um certificador da compositividade de n.

O Algoritmo 3 representa uma rodada do teste de Frobenius quadrático simplificado.

Algoritmo 3 Rodada do TFQS

Input: Inteiro ímpar n, inteiro pequeno c tal que $(c, n) = -1$ e o polinômio $\epsilon \in R(n, c)$ com $\epsilon^4 = -1$.

Output: Ou PRIMO ou COMPOSTO

1: Escolha z aleatório $\in R(n, c)$ com $(N(z)/n) = -1$
2: **if** $z^n \neq \bar{z}$ **then**
3: **return** COMPOSTO
4: **end if**
5: **if** $z^{\frac{n^2-1}{8}} \notin \{\pm\epsilon, \pm\epsilon^3\}$ **then**
6: **return** COMPOSTO
7: **end if**
8: **return** PRIMO

3. Implementação

O planejamento da implementação foi realizado sobre a biblioteca criptográfica RELIC (Aranha and Gouvêa). Para uma implementação completa do teste, é necessário uma ferramenta que forneça aritmética multi-precisão, suporte à manipulação de números inteiros muito grandes que extrapolariam a precisão de qualquer tipo primitivo de variáveis. Em resolução com a RELIC, a implementação foi executada utilizando a linguagem C padrão C99. Antes da implementação do teste em si, foi necessário elaborar funções específicas de utilidade comum e de operações em anel polinomial que não eram cobertas pela biblioteca. As principais delas serão discutidas nos tópicos a seguir.

3.1. Quadrado Perfeito

A raiz quadrada de um inteiro a com n *bits* deve estar contida no intervalo $(\ 2^{\lfloor \frac{n-1}{2} \rfloor},\ 2^{\lfloor \frac{n}{2} \rfloor})$. É possível, de forma semelhante a uma busca binária, encontrar a raiz inteira de a com o custo de $O(\log n)$ multiplicações. Tendo a raiz inteira c de a, basta verificar se $c^2 = a$ para determinar se a é quadrado perfeito.

3.2. Aritmética Básica

Adição e subtração de polinômios de mesmo grau é feita de maneira natural, respeitando a ordem de cada coeficiente e a redução módulo n. A multiplicação de dois polinômios de graus m e n, resulta em um polinômio de grau $m + n$. A aritmética do teste é realizada no grupo multiplicativo $R(n, c)^*$, módulo $\mathbb{Z}_n$ e o polinômio quadrático

$x^2 - c$. Assim, pode-se chegar a uma fórmula fechada para a multiplicação e redução modular.

Seja $a = a_1x + a_0$ e $b = b_1x + b_0$ polinômios com coeficientes em $\mathbb{Z}_n$, a multiplicação $a \cdot b$ em $R(n, c)$ tem resultado: $(a_1b_0 + a_0b_1)x + a_0b_0 + a_1b_1c$.

O custo de processamento desses algoritmos é limitado pelo número de multiplicações, quadrados e reduções modulares entre números grandes, portanto iremos desconsiderar o custo das somas e subtrações, que em geral são desprezíveis. Deste modo, uma multiplicação de dois polinômios tem o custo de quatro multiplicações e uma redução modular de coeficientes em $\mathbb{Z}_n$. Para o quadrado de um polinômio em $R(n, c)^*$, basta usar o mesmo resultado de anterior para $a = b$, que resulta em $a^2 = 2a_1a_0x + a_0^2 + a_1^2c$ com o custo de uma multiplicação e dois quadrados.

3.3. Exponenciação Modular

A maneira mais ingênua, até mesmo para números de precisão simples, de calcular $c = a^k \bmod n$ consiste em realizar $k - 1$ multiplicações. Em aplicações criptográficas, a grandeza de k normalmente excede 2^{512} ou 2^{1024}. Computar tais multiplicações tomaria mais tempo do que a vida do Universo, o que torna ineficaz tal abordagem. Podemos descrever a exponenciação na seguinte forma recursiva:

$$a^k = \begin{cases} 1 & \text{se } k = 0 \\ a \cdot \left(a^{\frac{k-1}{2}}\right)^2 & \text{se } k \equiv 1 \pmod 2 \\ \left(a^{\frac{k}{2}}\right)^2 & \text{se } k \equiv 0 \pmod 2 \end{cases}$$

Utilizando essa ideia, podemos entender as sucessivas divisões por 2 como um deslocamento à direita nos *bits* do expoente. O algoritmo implementado é o de exponenciação modular esquerda para direita, neste caso, a exponenciação modular é realizada com $O(\log n)$ quadrados e $o(\log n)$ multiplicações.

4. Otimização

Nesta Seção, serão descritas otimizações utilizadas na implementação eficiente do algoritmo TFQS.

4.1. Crivo para Quadrados Perfeitos

É possível chegar ao resultado de que todo quadrado perfeito par é múltiplo de 4, todo quadrado perfeito ímpar deixa resto 1 mod 4 e, por transitividade, 1 mod 8. Analisando o dígito menos significativo, e o seu quadrado, dos múltiplos de 4 e dos ímpares que deixam resto 1 na divisão por 4, chega-se a conclusão que os quadrados perfeitos só podem ter os algarismos $0, 1, 4, 5, 6$ e 9 como dígito menos significativo.

A partir dessa ideia, pode-se criar um crivo inicial que retorna FALSO para qualquer candidato a quadrado perfeito que termine em $2, 3, 7$ ou 8.

4.2. Multiplicação de Karatsuba

Anatolii A. Karatsuba, quando jovem aluno de graduação, propôs um eficiente algoritmo para multiplicação (Karatsuba and Ofman 1963) de dois números grandes que

contradizia uma conjectura lançada por seu professor, A. Kolmogorov, a respeito de um limite inferior para essa multiplicação. A mesma ideia de Karatsuba pode ser aplicada na multiplicação de polinômios: $a \cdot b = \big[(a_1 + a_0)(b_1 + b_0) - a_1b_1 - a_0b_0\big]x + a_0b_0 + a_1b_1c$.

Expandindo essa expressão de Karatsuba, pode-se notar que retorna ao mesmo resultado da multiplicação em 3.2. Assim, reutilizando os produtos já computados, o custo da multiplicação de dois polinômios cai para três multiplicações e uma redução modular.

4.3. Fórmula do Quadrado Complexo

Também se tem uma expressão utilizada no quadrado de número complexos, $(a + bi)$, que pode ser usada no quadrado de polinômios quadráticos: $a^2 = 2a_0a_1x + (a_0 + a_1)(a_0 + a_1c) - a_0a_1 - a_0a_1c$. O custo é reduzido para apenas duas multiplicações: $(a_0 + a_1)(a_0 + a_1c)$ e a_0a_1.

4.4. Janela Deslizante

O método janela deslizante para exponenciação modular implementado é baseado em uma generalização do algoritmo de exponenciação esquerda para direita que possibilita o processamento de mais de um *bit* do expoente por iteração através do pré-cálculo de exponenciações em uma janela. A janela deslizante consegue reduzir o pré-cálculo e a quantidade de multiplicações (Menezes et al. 1996).

4.5. Redução de Montgomery

Durante a exponenciação modular, são realizadas sucessivas multiplicações e quadrados. As repetidas reduções modulares em $\mathbb{Z}_n$ nessas operações é um fator limitante crítico do desempenho da função. Uma solução proposta por Montgomery é trocar as divisões por multiplicações. Contudo, há um detalhe importante: para ser realizada a redução de Montgomery é necessário o pré-cálculo de uma inversão em $\mathbb{Z}_n$, fato que deve ser considerado na análise do desempenho.

A otimização é alcançada quando é necessário uma repetição conhecida de operações que utilizam essa redução. Neste caso, a transformação na forma de Montgomery pode ser antecipada e a volta, atrasada para depois das repetições, aumentando o desempenho nas reduções modulares.

4.6. Manipulação Algébrica

4.6.1. Multiplicação por c pequeno

Analisando o Algoritmo 2 (MR2), vê-se que o inteiro retornado c varia em apenas três casos: $c = -1 \equiv n - 1$, $c = 2$ e $c =$ valor aleatório pequeno tal que $(c/n) = -1$. As multiplicações de inteiros multi-precisão por c podem ser simplificadas para todos os casos. No primeiro caso, podemos trocar uma multiplicação de $n - 1$ e uma redução módulo n por apenas uma subtração de n. No segundo caso, trocamos uma multiplicação e uma redução modular por um deslocamento à esquerda e uma subtração. Para o terceiro caso, podemos trocar a busca aleatória por uma busca incremental com no máximo 3 cálculos de Jacobi (Seysen 2005), assim mantendo o valor de c restritamente pequeno de modo que possamos substituir uma multiplicação e uma redução modular por uma multiplicação por dígito simples e $O(1)$ subtrações.

4.6.2. Reuso de uma exponenciação modular

É possível reduzir a quantidade de exponenciações modulares necessárias nos cálculos de z^n e $z^{(n^2-1)/8}$ no algoritmo (Seysen 2005). Tome $t = z^{\lfloor (n-1)/8 \rfloor} = z^{(n-1-\epsilon)/8}$, $\epsilon \in \mathbb{Z}$ e $0 \leq \epsilon < 8$. Computando t previamente e definindo ϵ, podemos calcular z^n com mais $O(1)$ multiplicações e quadrados. A partir de t, também podemos obter o valor de $z^{(n^2-1)/8}$ como segue: $z^{(n^2-1)/8} = z^{(n+1+\epsilon)(n-1-\epsilon)/8} \cdot z^{\epsilon(\epsilon+2)/8} = N(t) \cdot t^{\epsilon} \cdot z^{\epsilon(\epsilon+2)/8}$. Assim, de duas exponenciações modulares em anel polinomial é possível reduzir para apenas uma exponenciação e $O(1)$ multiplicações e quadrados.

4.7. Redução Preguiçosa

Entre as três multiplicações na implementação de Karatsuba é necessária uma redução modular extra para que não seja excedida a precisão da biblioteca, no caso dos operandos terem precisão máxima. A técnica da redução preguiçosa consiste em atrasar esta redução modular intermediária para o final da operação:

- Elimina-se a redução modular intermediária e define um fator de redução $r = n \cdot 2^{|n|}$.
- Quando necessária uma soma ou subtração de um produto com a precisão acumulada, realiza-se uma soma ou subtração de r como redução modular.

O custo final da multiplicação de Karatsuba em $R(n, c)$ é de três multiplicações e duas reduções modulares. Na operação de quadrado não há como usar a técnica da redução preguiçosa, pois o cenário ja é ótimo, duas multiplicações e duas reduções modulares.

5. Resultados

O ambiente de testes do projeto tem a seguinte configuração:

Tabela 1. Ambiente de testes.

Computador	Asus Q550L. Intel Core i7-4500U. 8GB DDR3.
OS / kernel	Ubuntu 13.10 x86˙64 / 3.11.0-15-generic
Compilador	gcc 4.8.1
RELIC	relic-0.3.5
GMP	2:5.1.2+dfsg amd64. *Multiprecision arithmetic library*

A cada passo de implementação e otimização dos algoritmos, foram realizados testes para garantir a corretude dos componentes e uma coleta de dados consistente. O artefato de testes foi desenvolvido em um ambiente fornecido pela própria biblioteca RELIC e verificava asssertivas sobre propriedades de anéis de polinômio. Após a implementação otimizada do TFQS, chegou-se à comparação com o teste de Miller-Rabin, algoritmo padrão da RELIC. Nesta análise, foram necessários alguns ajustes para igualar a parametrização dos algoritmos. Inicialmente, as rodadas do teste de Miller-Rabin estavam configuradas para uma probabilidade de erro de 2^{-80}, sendo necessário um ajuste no número de rodadas para se equiparar ao erro do TQFS, 2^{-100}. As Tabelas 2 e 3 apresentam alguns dos valores mais relevantes para o número de rodadas de cada algoritmo. O ajuste das rodadas do Miller-Rabin foi realizado através da ferramenta `mrtab` [1].

[1]Gerador de tabelas de limite de iteração para Miller-Rabin. `https://code.google.com/p/mrtab/`

Tabela 2. Probabilidade de erro 2^{-e} para o TFQS (*bits* k vs. número de rodadas t). Retirada de (Seysen 2005).

$k \setminus t$	1	2	3	4	5
300	48	100	124	143	160
400	57	118	146	169	189
500	65	134	166	192	214
600	72	148	184	212	237
1000	96	197	243	281	314

Tabela 3. Rodadas do Miller-Rabin ajustada para erro 2^{-100}.

t	1	2	3	4	5	6	7	8	9	10
k	3883	1761	1159	868	698	586	507	448	403	367

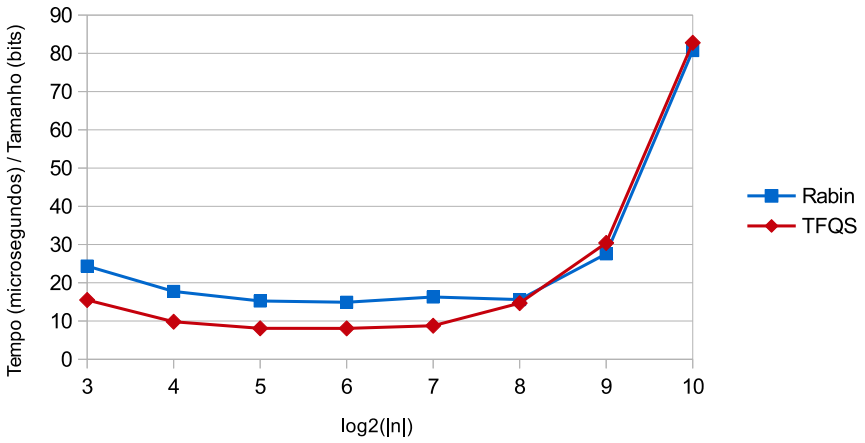


Figura 1. Gerador de primo com teste Miller-Rabin vs. TFQS.

Na comparação dos resultados obtidos (Figura 1), nota-se que o TFQS é mais eficiente para teste de inteiros de até 256 *bits*. Neste intervalo, obteve-se uma redução de cerca de 30% no tempo de execução. A partir de 512 *bits*, o Miller-Rabin passa a ser mais eficiente. Em uma visão panorâmica, o TFQS pode ser delimitado pelo custo de uma exponenciação modular em anel de polinômio por rodada. De maneira semelhante, o teste de Miller-Rabin pode ser delimitado por uma exponenciação modular simples por rodada.

6. Conclusão

Neste trabalho foi apresentada uma implementação eficiente de um teste de primalidade ainda novo que obteve resultado satisfatório para inteiros de até 256 *bits*. No seu cenário ótimo, o TFQS se mostrou, em média, 30% mais eficiente que o teste de Miller-Rabin. Também foram apresentados métodos de aritmética em anel de polinômio e técnicas de otimização que culminaram na produção de um artefato eficaz. Conclui-se que o TFQS possui iterações mais longas, no entanto, executa menos rodadas devido sua baixa probabilidade de erro. O teste de Miller-Rabin possui iterações mais rápidas, devido aritmética mais simples, mas executa maior número de rodadas. À medida que a precisão

do inteiro testado aumenta, essa vantagem do TFQS diminui, tornando nula quando o número de rodadas do Miller-Rabin iguala ao custo de uma exponenciação modular em anel de polinômio em relação ao custo de uma exponenciação modular simples.

O escopo principal do trabalho foi atingido parcialmente, houve a implementação de um teste mais eficiente, porém com restrição de magnitude. O problema de gerar chaves com alto nível de segurança em dispositivos com poder computacional limitado em tempo hábil ainda persiste.

Como trabalhos futuros, propõe-se o estudo de outros testes presentes na indústria como o teste de Lucas, Baillie-PSW (Baillie and Wagstaff 1980) e Miller-Rabin combinado ao teste de Lucas-Selfridge (Pomerance et al. 1980) que é baseado no teste Baillie-PWS e aperfeiçoado por Selfridge.

Por fim, todo código produzido nesta obra está disponível no repositório online (`https://code.google.com/p/bruno-tg2/`) com um arquivo *bash* que gerencia a configuração, compilação e execução de testes, *benchmarks*, produção de gráfico de desempenho, entre outros.

Referências

[Agrawal et al. 2004] Agrawal, M., Kayal, N., and Saxena, N. (2004). PRIMES is in P. *Annals of Mathematics*, 160(2):781–793.

[Aranha and Gouvêa] Aranha, D. F. and Gouvêa, C. P. L. RELIC is an Efficient Library for Cryptography. `http://code.google.com/p/relic-toolkit/`.

[Baillie and Wagstaff 1980] Baillie, R. and Wagstaff, S. (1980). Lucas pseudoprimes. *Mathematics of Computation*, 35(152):1391–1417.

[Damgard and Frandsen 2003] Damgard, I. B. and Frandsen, G. S. (2003). An extended quadratic frobenius primality testwith average an worst case error estimates.

[Dietzfelbinger 2004] Dietzfelbinger, M. (2004). *Primality Testing in Polynomial Time*. Lecture Notes in Computer Science. Springer.

[Grantham 1998] Grantham, J. (1998). A probable prime test with high confidence. *J. Number Theory*, 72(1):32–47.

[ISO/IEC 18032 2005] ISO/IEC 18032 (2005). Information technology - Security techniques - Prime number generation.

[Karatsuba and Ofman 1963] Karatsuba, A. and Ofman, Y. (1963). Multiplication of many-digital numbers by automatic computers. *Physics-Doklady*, 7:595–596.

[Menezes et al. 1996] Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC Press.

[Miller 1976] Miller, G. L. (1976). Riemann's hypothesis and tests for primality. *Journal of Computer and Systems Sciences*, 13(3):300–317.

[Müller 2001] Müller, S. (2001). A probable prime test with very high confidence for $n = 1 \bmod 4$. *Lecture Notes in Computer Science*, pages 87–106.

[Müller 2003] Müller, S. (2003). A probable prime test with very high confidence for $n = 3 \bmod 4$. *J. Cryptology*, 16(2):117–139.

[Pomerance et al. 1980] Pomerance, C., Selfridge, J. L., and Wagstaff, S. (1980). The pseudoprimes to $25 \cdot 10^9$. *Mathematics of Computation*, 35(151):1003–1026.

[Rabin 1980] Rabin, M. O. (1980). Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12(1):128–138.

[Seysen 2005] Seysen, M. (2005). A Simplified Quadratic Frobenius Primality Test.

Implementação do esquema totalmente homomórfico sobre inteiros de chave reduzida.

Guilherme Rodrigues Bilar, Luan Cardoso dos Santos, Fabio Dacêncio Pereira
COMPSI- Computing and Information Systems Research Lab – Centro Universitário Eurípedes de Marília
Avenida Higino Muzi Filho, 529 - Cep 17525-901 Marilia/SP
lcsantos.lcsantos@gmail.com; guibilar@gmail.com; prof.fabiopereira@gmail.com

Resumo: *Neste artigo é implementado o esquema totalmente homomórfico de chave reduzida (DGHV sobre inteiros) proposto por Jean-Sébastian Coron, Avradip Mandal, David Naccache e Mehdi Tibouchi, que foi publicado na conferencia CRYPTO 2011, este mesmo esquema pode ser comparado com o esquema totalmente homomórfico de Gentry, que se trata de um esquema totalmente homomórfico de construção mais simples, contudo essa simplicidade vem ao custo de que sua chave pública possui um tamanho estimado de* $\tilde{O}(\lambda^{10})$*, o que de acordo com Coron et al, torna inviável a aplicação em sistemas práticos. O esquema totalmente homomórfico DGHV com chave pública reduzida diminui o tamanho da chave pública gerada para* $\tilde{O}(\lambda^{7})$ *criptografando de maneira quadrática os elementos da chave pública, ao invés de criptografá-los de maneira linear. Para fazê-lo foi utilizada a linguagem de programação Python, contando com a biblioteca de matemática e teoria numérica GMPY2.*

Abstract: *In this paper we implemented the fully homomorphic scheme with shoter key (DGHV with shorter key) proposed by Jean-Sébastian Coron, Avradip Mandal , David Naccache and Mehdi Tibouchi , which was published in the conference CRYPTO 2011, this same scheme can be compared with the Gentry's fully homomorphic scheme, being it a fully homomorphic scheme with a simpler construction, however this simplicity comes at the cost of the public key having an estimated size of* $\tilde{O}(\lambda^{10})$*, which according to Coron et al, makes it impossible to use in practical systems. The DGHV scheme over integers with shorter public key decreases the size of the generated public key to* $\tilde{O}(\lambda^{7})$*, encrypting the information of the public key in a quadratic manner, instead of encrypting them in a linear way. To do so, the python programming language was used, with the library of mathematics and number theory GMPY2.*

1. Introdução

A criptografia moderna tem como base problemas matemáticos relacionados à teoria de números, exemplo deste é o algoritmo RSA que usa a fatoração de números inteiros em números primos para a geração de suas chaves assimétricas [Rivest et al., 1978], outro exemplo disso é o uso de logaritmos discretos na criptografia de curvas elípticas [Miller, 1985], entretanto, o surgimento da teoria dos computadores quântico e do algoritmo de Shor tornou este tipo de solução vulnerável a ataques quânticos [Shor, 1994]. Para proteger dados, até mesmo da criptoanálise quântica, tivemos o surgimento de uma nova área dentro da segurança da informação, chamada de Criptografia Pós-Quântica. Essa area estuda a criação de algoritmos criptográficos resistentes a ataques

quânticos, tendo como base outros problemas matemático-computacionais tais como as classes de algoritmos baseados em *hash*, os baseados em reticulados, os baseados em código, entre outros. O tempo de execução de ataque para todos esses algoritmos é exponencial, mesmo para um computador quântico [Bernstein et al. 2008].

Dentre os esquemas criptográficos pós-quânticos, destacam-se aqueles que possuem a característica de serem totalmente homomórficos. Um esquema é dito totalmente homomórfico quando se é possível avaliar circuitos lógicos de profundidade arbitrária nos dados cifrados, de forma pública [Gentry, 2009]. Em outras palavras, existe uma primitiva Eval, que dada uma função *f( )* e um texto cifrado *E(m)*, sendo *E( )* a função criptográfica e "*m*" a mensagem, retorna *E(f(m))*.

A figura I ilustra os principais tipos de esquemas totalmente homomórficos existentes. O primeiro esquema totalmente homomórfico conhecido foi o proposto por Gentry em sua tese original, proposta em 2009, o mesmo incluía pela primeira vez a primitiva *recrypt*, que dava ao esquema, originalmente parcialmente homomórfico, a característica de ser totalmente homomórfico, que utiliza como base reticulados ideais. Estudando a tese de Gentry, e seu esquema de *bootstraping* como base, Halevi, em parceria com o próprio Gentry, implementaram em 2010, o esquema totalmente homomórfico Gentry-Halevi, este basicamente o esquema original de Gentry, com inclusão de algumas melhorias.

Ainda em 2010, Dijk, Gentry, Halevi e Vaikuntanathan (DGHV), propuseram um esquema totalmente homomórfico que utiliza apenas álgebra modular sobre o anel de números inteiros. Este mesmo esquema foi posteriormente analisado por Coron em 2011, que propôs duas técnicas de redução de chave pública para o esquema, dando origem a duas variantes do esquema DGHV. A primeira variante de Coron, chamada de DGHV com chave reduzida, conta com a adição de novos parâmetros quadráticos adicionados as primitivas do esquema, armazenando apenas um pequeno conjunto da chave pública para então gerar a chave pública completa em tempo de execução. A segunda variante de Coron, chamada de DGHV com compactação de chave pública com troca de modulo, utiliza geradores de números pseudoaleatórios e armazena apenas as sementes e os fatores de correção para os números pseudoaleatórios, recuperando os valores em tempo de execução.

Em 2011, Brakerski e Vaikuntanathan, propuseram um novo tipo de esquema totalmente homomórfico, este é baseado no problema de aprendizagem com erros (sigla em inglês: LWE), este em especifico aplicando a problemas LWE em anel (da sigla em inglês: RLWE), este aplica os resultados conhecidos para o RLWE diretamente no esquema. Já em 2012, Brakerski, Gentry e Vaikuntanathan, apresentaram um novo esquema totalmente homomórfico com base no problema LWE, este esquema funciona sem a necessidade de operação de *bootstrapping*, porém mantendo-a no esquema como forma de melhorar a eficiência.

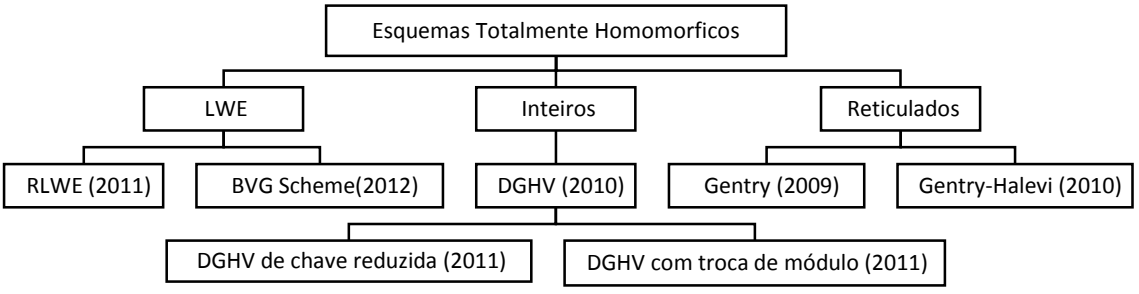


Figura I - Principais esquemas FHE

Baseado no trabalho de Gentry, dois tipos de esquemas homomórficos são conhecidos: O esquema de Gentry com reticulados ideias [Gentry, 2009] e o esquema DGHV sobre inteiros. O uso de reticulados na implementação não se mostrou uma das melhores alternativas, de acordo com Coron et al [Coron et al., 2011] e de Gentry [Dijk et al., 2010], pois o tamanho das chaves públicas geradas era na ordem de $\tilde{\mathcal{O}}(\lambda^{10})$, com a implementação de Coron [Coron et al., 2011], na qual foram utilizados números inteiros na geração de chaves, obteve-se uma redução para um tamanho estimado de $\tilde{\mathcal{O}}(\lambda^{7})$ da chave pública. O presente trabalho tem como proposta implementar o esquema totalmente homomórfico sobre inteiros de chave reduzida proposto por Coron, [Coron et al., 2011].

2. Fundamentação teórica

O esquema implementado neste artigo é uma variante proposta por Coron do esquema originalmente proposto por Dijk, Gentry, Halevi e Vaikuntanathan (DGHV) no Eurocrypto 2010 [Dijk et al., 2010].

2.1. DGHV sobre inteiros

Segue uma breve descrição do sistema original sobre números inteiros, utilizado por Coron como base para seu trabalho. Gentry define que DGHV sobre inteiros utiliza como base um conjunto de inteiros públicos, $x_i = p.q_i + r_i, 0 \leq i \leq \tau$, onde o inteiro p é secreto [Gentry e Halevi, 2011], sendo dado um parâmetro de segurança λ, os seguintes parâmetros devem ser utilizados para compor o esquema SHE (do inglês, encriptação parcialmente homomórfica), que gera o FHE (do inglês, encriptação totalmente homomórfica) sobre inteiros:

- γ é o comprimento em bits de x_i's;
- η é o comprimento em bits da chave secreta p;
- ρ é o comprimento em bits do ruído r_i;
- τ é o número de x_i's na chave pública;
- ρ'é um parâmetro de ruído secundário utilizado para cifrar.

Que devem seguir as seguintes restrições, de forma a mitigar possíveis ataques:

- $\rho = \omega(log\lambda)$, para proteção contra ataques de força bruta direcionados ao ruído;
- $\eta \geq \rho.\Theta(\lambda log^2\lambda)$ para que seja possível realizar operações homomórficos para avaliar o "circuito de decriptação reduzido";
- $\gamma = \omega(\eta^2.log\lambda)$ para frustrar ataques baseados em retículos com aproximação pelo problema de MDC;
- $\tau \geq \gamma + \omega(log\lambda)$ para reduzir a aproximação por MDC;
- $\rho' = \rho + \omega(log\lambda)$ para o parâmetro de ruído secundário.

Dados esses parâmetros é possível gerar as primitivas do esquema DGHV, sendo que para um inteiro ímpar p de η-bit, seja utilizada uma distribuição sobre inteiros de γ-bit:

$$\mathcal{D}_{\gamma,\rho}(p) = \{ Escolha\ q\ \leftarrow \mathbb{Z} \cap \left[0, \frac{2^{\gamma}}{p}\right), r \leftarrow \mathbb{Z} \cap (-2^{\rho}, 2^{\rho}) : Saída\ x = q.p + r\}$$

2.1.1. Primitivas Do DGHV

As primitivas deste esquema são quatro, *KeyGen, Encrypt, Decrypt* e *Evaluate*. *KeyGen* é a primitiva responsável pela geração do par de chaves do esquema, *Encrypt* responsável por gerar o texto cifrado, *Decrypt* responsável por decifrar o texto cifrado, e *Evaluate*, que executa, de forma pública, um circuito lógico sobre uma tupla de bits cifrados, e, que retorna o equivalente cifrado desse circuito aplicado aos dados originais. Sendo assim obtemos as primitivas como sendo:

i. *KeyGen*(λ):

Sendo a chave privada um inteiro ímpar de η-bits, $p \leftarrow (2\mathbb{Z}+1) \cap [2^{\eta-1}, 2^{\eta})$, para gerar a chave pública amostramos $x_i \leftarrow \mathcal{D}_{\gamma,\rho}(p)$ para $i = 0, \dots, \tau$, reordenando de forma que x_0 seja maior o maior elemento do conjunto, e recomeçando o processo a não ser que x_0 seja um número ímpar e $[x_0]_p$ seja par. Assim a chave pública é definida como pk=$\langle x_0, x_1, \dots, x_\tau \rangle$.

ii. *Encrypt* (pk, $m \in \{0,1\}$):

Escolhe-se um subconjunto $S \subseteq \{1,2,\dots,\tau\}$ aleatório, e um inteiro aleatório *r* entre $\left(-2^{\rho'}, 2^{\rho'}\right)$, e gera o texto cifrado *c* como:

$$c = \left[m + 2r + 2\sum_{i \in S} x_i \right]_{x0}$$

iii. *Evaluate*(pk, $C, c_1, \dots, c_t$):

Dado o circuito C com *t* bits de entrada e *t* textos cifrados c_i, aplicar a adição e multiplicação das portas lógicas de C nos textos cifrados, executando todas as operações sobre os inteiros, e retornar o inteiro resultante.

iv. *Decrypt*(sk, *c*):

Tem como saída $m \leftarrow (c \bmod p) \bmod 2$. Note que $c \bmod p = c - p\,.\lfloor c/p \rceil$ e p é ímpar, então a função *decrypt* pode ser calculada como: $m \leftarrow [c]_2 \oplus [\lfloor c/p \rceil]_2$.

Assim é descrito o esquema criptográfico parcialmente homomórfico, sendo este semanticamente seguro contra ataques de aproximação de MDC.

2.2. DGHV Sobre Inteiros Com Chave Pública Reduzida

Coron [Coron et.al., 2011] utilizou uma variante do esquema DGHV, onde foi adicionado um novo parâmetro β, sendo manipulado números inteiros $x'_{i,j}$, na forma de $x'_{i,j} = x_{i,0}.x_{j,1} \bmod x_0$ para $1 \le i,j \le \beta$. Dessa forma, apenas 2β inteiros precisam ser armazenados para gerar os $\tau = \beta^2$ inteiros utilizados para a criptografia. Em outras palavras, o sistema executa a criptografia utilizando elementos na forma quadrática da chave pública, ao contrario da forma linear adotada anteriormente. Tal permite reduzir o tamanho da chave pública de τ para $2\sqrt{\tau}$ inteiros de γ bits [Coron et al, 2011].

Além disso, a fim de tornar esse esquema totalmente homomórfico, são necessários λ^3 elementos y com comprimento de $\kappa = \gamma + 2 + \lceil \log_2(\lambda+1) \rceil$, o que aumentaria o tamanho da chave pública de $\tilde{\mathcal{O}}(\lambda^7)$ para $\tilde{\mathcal{O}}(\lambda^8)$. Coron então propôs que apenas o primeiro dos elementos de y fosse armazenado na chave, e, os outros gerados a partir de uma função geradora de números aleatórios. Dessa forma, o tamanho da chave é mantido na ordem de λ^7, e, os elementos de y são recuperados em tempo de execução [Coron et al, 2011].

É apresentada a seguir a descrição completa do esquema proposto por Coron:

i. *KeyGen*(1^λ):

Gera um número ímpar p com η bits. Escolhe um número inteiro $q_0 \in [0, 2^\gamma / p)$, escolhido como um produto de dois primos aleatórios de λ^2 bits, e $x_0 = q_0 \cdot p$. Gera β pares inteiros de $x_{i,0}, x_{i,1}$ dentro do intervalo de $1 \leq i \leq \beta$:

$$x_{i,b} = p.q_{i,b} + r_{i,b}, 1 \leq i \leq \beta, 0 \leq b \leq 1$$

onde, $r_{i,b}$ são inteiros entre $(-2^\rho, 2^\rho)$ e $q_{i,b}$ são inteiros aleatórios de ordem $[0, q_0)$. Sendo assim $pk^* = (x_0, x_{1,0}, x_{1,1}, \dots, x_{\beta,0}, x_{\beta,1})$.

Também gera os vetores $s^{(0)}$ e $s^{(1)}$, de comprimento $\lceil\sqrt{\Theta}\rceil$, que seguem a condição de que $s_1^{(0)} = s_1^{(1)} = 1$, para cada um dos $\kappa \in [0, \sqrt{\theta})$ e $b = 0,1$, onde há pelo menos um bit não zero entre os $s_i^{(b)}$'s, $k\lfloor\sqrt{B}\rfloor < i \leq (k+1)\lfloor\sqrt{B}\rfloor$, com $B = \Theta/\theta$, e S sendo $S = \{(i,j)\colon s_i^{(0)}.s_j^{(1)} = 1\}$, contendo exatamente θ elementos.

Inicializa um gerador f de números pseudoaleatórios válido para todo o sistema com uma semente aleatória se, usando assim *f*(se) para gerar $u_{i,j} \in [0, 2^{\kappa+1})$ para $1 \leq i, j \leq \lceil\sqrt{\Theta}\rceil, (i,j) \neq (1,1)$. Então, atribuindo $u_{1,1}$ de forma que:

$$\sum_{(i,j)\in S} u_{i,j} = x_p \; mod \; 2^{\kappa+1}$$

onde $x_p \leftarrow \lfloor 2^\kappa / p \rceil$.

Assim, computando a cifra de $\sigma^{(b)}$dos vetores $s^{(b)}$, escolhendo para cada $i \in [1, \lceil\sqrt{\Theta}\rceil]$ e $b = 0,1$, inteiros aleatórios $r'_{i,b} \in (-2^\rho, 2^\rho)$ e $q'_{i,b} \in [0, q_0)$, é determinado que:

$$\sigma_i^{(b)} = s_i^{(b)} + 2r'_{i,b} + p.q'_{i,b} \bmod x_0$$

Deste modo, temos como saída da primitiva KeyGen(), a chave privada sendo $sk = (s^{(0)}, s^{(1)})$, e a chave pública como $pk = \left(pk^*, \text{se}, u_{1,1}, \sigma^{(0)}, \sigma^{(1)}\right)$.

ii. *Encrypt*(*pk*, $m \in \{0,1\}$):

Escolha uma matriz de números aleatórios $b = (b_{i,j})_{1\leq i,j\leq\beta} \in [0,2^\alpha)^{\beta\times\beta}$ e um inteiro aleatório *r* no intervalo $(-2^{\rho'}, 2^{\rho'})$. Retorne o texto cifrado como:

$$c^* = m + 2r + 2 \sum_{1\leq i,j\leq\beta} b_{ij} \cdot x_{i,0} \cdot x_{j,1} \; mod \; x_0$$

iii. *Add*(pk, c_1^*, c_2^*): Retorna $c_1^* + c_2^* \; mod \; x_0$
iv. *Mult*(pk, c_1^*, c_2^*): Retorna $c_1^* \cdot c_2^* \; mod \; x_0$
v. *Expand*(pk, c^*):

Esse procedimento, a expansão do texto cifrado, recebe um texto cifrado c^* e computa a matriz associada *z*. Podemos pensar nesse procedimento como separado tanto do *Encrypt* quanto de *Decrypt*, já que pode ser executado de forma pública utilizando apenas o texto cifrado e dados públicos. Para tal, para cada $1 < i, j < \sqrt{\Theta}$, primeiramente compute o número inteiro aleatório $u_{i,j}$ usando o gerador de números pseudoaleatórios *f*(se), então calcule $y_{i,j} = u_{i,j}/2^\kappa$ e então compute $z_{i,j}$:

$$z_{i,j} = [c^* \cdot y_{i,j}]_2$$

mantendo apenas $n = \lceil log_2(\theta + 1)\rceil$ bits de precisão após o ponto binário. Defina a matriz $z = \left(z_{i,j}\right)$. Retorne o texto cifrado expandido $c = (c^*, z)$

vi. *Decrypt* (sk, c^*, z):Retorna $m \leftarrow \left[c^* - \left\lfloor \sum_{i,j} s_i^{(0)} \cdot s_j^{(1)} \cdot z_{ij} \right\rfloor\right]_2$.
vii. *Recrypt*(pk, c^*, z):

Aplique o circuito de decriptação ao texto cifrado expandido Z e as chaves secretas encriptadas $\sigma_i^{(b)}$. Retorne o resultado como o texto cifrado renovado c^*_{new}.

Assim finalizamos a descrição completa do esquema DGHV Totalmente Homomórfico de chave pública reduzida, a primitiva *Evaluate*, foi fragmentada, dando origem a outras a outras duas primitivas de nome *Add* e *Mult*, que emulam, respectivamente, o comportamento das portas lógicas *XOR* e *AND* aplicadas ao texto puro, de forma bit-a-bit. Tal esquema é seguro contra ataques baseados em MDC, como foi provado na sessão 4 da publicação que descreve essa variante do esquema [Coron et al, 2011].

3. Implementação

Para a implementação desse esquema, foi utilizada a linguagem Python 3.3, com a biblioteca GMPY2 para cálculos numéricos.

Inicialmente, foi criado um módulo, *fheKey.py*, que é responsável pelo processo de escrita e leitura das classes das chaves:

```
1   import pickle
2   class pk():
3   '''clase para armazenar a chave privada'''
4   def __init__(self, pkAsk, se, u11, sigma0, sigma1, P):
5   self.pkAsk=pkAsk
6   self.se=se
7   self.u11=u11
8   self.sigma0=sigma0
9   self.sigma1=sigma1
10  self.P=P
11  class sk():
12  '''classe para armazenar a chave pública'''
13  def __init__(self, sz, su, P):
14  self.s0=sz
15  self.s1=su
16  self.P=P
18  def write(obj, name):
20  with open(name, 'wb') as arq:
21  pickle.dump(obj, arq)
22  def read(name):
24  with open(name, 'rb') as arq:
25  return pickle.load(arq)
```

As classes *pk* e *sk* funcionam como contêineres para os parâmetros da chave pública e da chave secreta, respectivamente, assim como dos parâmetros concretos. As funções *write*() e *read*() encapsulam o módulo *pickle* do Python, que executa a escrita e leitura de objetos do Python em arquivos binários.

Também foi criada uma classe para encapsular os parâmetros da criptografia, conforme descritos experimentalmente pro Gentry:

Para executar a geração do par de chaves criptográficas, é utilizada a seguinte função:

```
1   def keygen(file, size='toy'):
2       P.setPar(size)
3       tempo=-time.time()
```

```
5       p = genZi(P._eta)
6       q0, x0 = genX0(p, P._gamma, P._lambda)
7       listaX = genX(P._beta, P._rho, p, q0)
8       pkAsk = listaX
9       pkAsk.insert(0, x0)
10      while True:
11          s0,s1=genSk(P._theta, P._thetam)
12          if (s0.count(1)*s1.count(1)==15): break
14      se=int(time.time()*1000) #seed para RNG
15      _kappa=P._gamma+6
16      u11=genU11(se, s0, s1, P._theta, _kappa, p)
17      sigma0 = encryptVector(s0, p, q0, x0, P._rho)
18      sigma1 = encryptVector(s1, p, q0, x0, P._rho)
19      tempo+=time.time()
20      #picle files
21      public=fheKey.pk(pkAsk, se, u11, sigma0, sigma1, P)
22      secret=fheKey.sk(s0, s1, P)
23      fheKey.write(public, 'pk_pickle_'+file)
24      fheKey.write(secret, 'sk_pickle_'+file)
```

A função *keygen* recebe como parâmetros o nome básico para os arquivos onde serão escritos a chave, e, opcionalmente o parâmetro de segurança que será utilizado. Caso esse parâmetro seja omitido, a função executará no padrão menos seguro.

Primeiramente a função gera um número inteiro ímpar P de η bits, os valores q0 e x0, assim como uma lista de valores aleatórios xi. Em seguida, a função gera a chave pública parcial pk* (linhas 5-13). Em seguida, são gerados os vetores de bits aleatórios, e verificado se o peso de Hamming de $s^1 \times s^0 = \theta$ (linha 14).

A função então utiliza o tempo local, em milissegundos, como semente para o RNG(Gerador de números pseudoaleatórios). Utilizando o RNG do Python, é gerada uma matriz de valores U a partir dos vetores s^b. Com essa matriz é calculado o elemento u_{11}(linha 20). Por ultimo, são calculados os vetores σ^b, que são encriptações de s^b. Por fim, a função cria objetos da classe *fheKey.pk* e *fheKey.sk*, e os escreve no disco. Também é gerado um arquivo em texto puro dos valores computados que é utilizado para verificação.

Função de Encrypt:

```
1   def encrypt(pk, m):
2       if m != 0 and m != 1:
3           print('not a valid m')
4           return 0
5
6       alpha = pk.P._lambda
        matrix = [[0 for i in range(pk.P._beta)] for j in
7   range(pk.P._beta)]
8       for i in range(pk.P._beta):
9           for j in range(pk.P._beta):
                matrix[i][j]=random.randint(0, 2**alpha -
10  1)
```

```
11          pprime=2**pk.P._rho
12          r=random.randint(-pprime, pprime)
13          pkAsk=(pk.pkAsk).copy()
14          x0=pkAsk.pop(0)
15          xi0=pkAsk[::2]
16          xj1=pkAsk[1::2]
17          ## iniciando processo de encrypt
18          somatorio=mpz(0)
19          for i in range(pk.P._beta):
20              for j in range(pk.P._beta):
22                  x=gmpy2.mul(xj1[j], xi0[i])
23                  somatorio += gmpy2.mul(matrix[i][j], x)
24          c=(mpz(m)+2*r+2*somatorio)%x0
25          return c
```

A função de *encrypt* recebe como parâmetros a chave pública, e o bit a ser criptografado. Primeiramente, é criada uma matriz de números aleatórios $\beta \times \beta$ e populada com valores no intervalo de (0, 2^{α}] (linha 8). Então, é definido um número inteiro aleatório no intervalo de (-2^{ρ}, 2^{ρ}). Por ultimo a função calcula o texto cifrado $c = m + 2r + 2\sum_{1 \le i,j \le \beta} b_{ij} \cdot x_{i,0} \cdot x_{j,1} \; mod \; x_0$. O valor de c é retornado como um inteiro de precisão múltipla.

Função de expansão:

```
1   def expand(pk, cAsk):
2       #cria uma matriz de sqrt(theta) elementos
3       l=int(math.sqrt(pk.P._theta))
4       y=[[0 for i in range(l)] for i in range(l)]
5       kappa=pk.P._gamma+6
6       n=2 ##ceil(log2(thetaM+1))
7       gmpy2.get_context().precision=n
8       for i in range(l): #computa matriz y[i][j]
9           for j in range(l):
                y[i][j]=float(gmpy2.mpfr(randomMatrix(i,
10  j, kappa, pk.se, l)/(2**kappa)))
11      y[0][0]=float(gmpy2.mpfr(pk.u11/2**kappa))
12      gmpy2.get_context().precision=21000
13      expand = [[1 for i in range(l)] for i in range(l)]
14      for i in range(l):
15          for j in range(l):
16              expand[i][j]=float((cAsk * y[i][j])%2)
17      return expand
```

A função de expansão recebe como parâmetros a chave pública, e a cifra c, e, a partir deles, computa a matriz associada *z*. Essa função é separada das primitivas *encrypt* e *decrypt*, já que pode ser computada de forma pública a partir apenas de dados públicos. Para tal, é criada uma matriz de y $\lceil\sqrt{\Theta}\rceil$ elementos, que é populada usando o RNG(se). Então, a matriz z é calculada como $z_{i,j} = [c^* \cdot y_{i,j}]_2$.

Função de decriptação:

```
def decrypt(sk, cAsk, z):
    soma=0
    l=int(math.sqrt(sk.P._theta))
    for i in range(l):
        for j in range(l):
            soma+=(sk.s0[i]*sk.s1[j]*z[i][j])
    soma=mpz(gmpy2.round_away(soma))
    m=(cAsk-soma)%2
    return m
```

A função de decriptação recebe como parâmetros a chave secreta, a mensagem cifrada e a matriz z. Então, a função calcula $soma = \left\lfloor \sum_{i,j} s_i^{(0)} \cdot s_j^{(1)} \cdot z_{i,j} \right\rceil$ e retorna o bit decriptado como $[c^* - soma]_2$.

Funções de cálculo homomórfico:

```
def add(pk, c1, c2):
    soma=gmpy2.add(c1,c2)
    return soma%pk.pkAsk[0]
def mul(pk, c1, c2):
    mul= gmpy2.mul(c1,c2)
    return mul%pk.pkAsk[0]
```

Essas funções são o equivalente as portas lógicas *XOR* e *AND* para os bits do texto cifrado. *XOR* é definido como uma soma de dois textos cifrados módulo x0 e *AND* é definido como uma multiplicação de dois textos cifrados módulo x0.

Tabela 1. Tempos de execução

Parâmetro	*KeyGen*	*Encrypt*	*Decrypt*	*Expand*
Toy	0.6 s	0.00236 s	0.0001 s	1.28103 s
Small	10 s	0.01294 s	0.0002 s	8.08505 s
Medium	1 min 1 s	*	*	*
Large	11 min 21s	*	*	*

(*estouro de memória)

Os tempos de execução obtidos (ver tabela 1) foram gerados em um único núcleo de um processador Core i5 m520 com 2.40Ghz de frequência, em um computador com sistema operacional de 64 bits e 4 GB de memória.

4. Conclusão

A escolha da implementação em linguagem Python trouxe algumas facilidades na geração de código. Por exemplo, operações com listas tornam algumas tarefas extremamente simples de se realizar: Na geração de pares s para a função de *keygen*, um dos primeiros passos é gerar uma lista de tamanho $\sqrt{\Theta}$, inicialmente com todos os elementos iguais a zero e o primeiro elemento igual a um, sendo que essa tarefa é trivialmente executada em Python.

A maior dificuldade encontrada durante esse projeto foi, inicialmente, conseguir informações sobre os esquemas homomórficos e, transformar e interpretar as ideias, apresentadas quase sempre como fórmulas e abstrações matemáticas em códigos de

computador. Também foi desafiador compilar o trabalho original de Gentry, já que exigem procedimentos diferentes e específicos para a correta compilação do código.

Junto das facilidades inerentes a linguagem, as operações matemáticas de alto desempenho e as primitivas necessárias para números inteiros de precisão múltipla foram providas pela biblioteca de teoria numérica GMPY2. Tal biblioteca, além de disponibilizar uma primitiva de números inteiros de múltipla precisão, necessário para os cálculos homomórficos, permite que os cálculos com esses números sejam notavelmente mais rápidos que as primitivas disponibilizadas pelo Python.

Além disso, a programação a partir da descrição matemática se mostrou uma excelente maneira de se compreender o conceito e a matemática por trás do homomorfismo, permitindo dessa forma adquirir um conhecimento mais profundo sobre o funcionamento do esquema homomórfico.

A próxima etapa desse trabalho será adicionar a função de *recrypt*, necessária para se executar circuitos de profundidade arbitrária no *cipherthext*. Posteriormente, esse código será modificado para ser executado de forma paralela em GPGPUS, utilizando-se de recursos tais como pyCuda, e NumbaPro.

Referências bibliográficas

RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. M. A method for obtaining digital signatures and public-key cryptosystems. *Commun.* ACM, 21(2):120–126, 1978.

MILLER, V. Uses of elliptic curves in cryptography. *In Advances in Cryptology*, Crypto 85, Lecture Notes in Computer Science, pages 417–426. Springer, 1985
SHOR, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In Shor, P. W., editor, *35th Annual Symposium on Foundations of Computer Science* (Santa Fe, NM, 1994), p. 124–134. IEEE Comput. Soc. Press, 1994.

BERNSTEIN, D. J., BUCHMANN, J. A., DAHMEN E. Post-Quantum Cryptography, Chicago and Darmstadt, 2008.

GENTRY, C.,A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University, 2009, Disponivel em:http://crypto.stanford.edu/craig.

CORON, J.S., MANDAL, A., NACCACHE, D. e TIBOUCHI, M., Fully Homomorphic Encryption over the Integers with Shorter Public Keys. In P. Rogaway (Ed.), *CRYPTO 2011, LNCS, vol. 6841*, Springer, pp. 487-504. Versão complete disponível em IACR eprint, 2011.

DIJK., M. VAN, GENTRY, C., HALEVI, S. e VAIKUNTANATHAN, V., Fully homomorphic encryption over the integers. *In H. Gilbert* (Ed.), EUROCRYPT 2010, LNCS, vol. 6110, Springer, p. 24-43, 2010.

GENTRY, C., e HALEVI, S., "Implementing Gentry's fully-homomorphic encryption scheme," *Advances in Cryptology*-EUROCRYPT 2011, p. 129-148, 2011.

GENTRY, C., Fully Homomorphic Encryption Using Ideal Lattices, Symposium on Theory of Computing - STOC, p.169-178, 2009.

Análise dos Desafios para Estabelecer e Manter Sistema de Gestão de Segurança da Informação no Cenário Brasileiro

Rodrigo Valle Fazenda[1], Leonardo Lemes Fagundes[2]

Instituto de Informática – Universidade do Vale do Rio dos Sinos (UNISINOS)
Caixa Postal 275 – CEP 93022-000 – São Leopoldo – RS – Brasil

Graduação Tecnológica em Segurança da Informação

[1]FazendaRodrigoV@gmail.com, [2]Llemes@unisinos.br

Resumo. *O estabelecimento da norma ISO 27001 cresce entre as organizações em todo o mundo. Porém, desafios são enfrentados pelas empresas para implementar esta norma. É escassa a quantidade de estudos sobre os desafios que empresas brasileiras enfrentam para estabelecer e manter o Sistema de Gestão de Segurança da Informação. Este artigo tem como objetivo identificar e analisar os desafios enfrentados para estabelecer e manter este sistema de gestão no cenário nacional. Através do método de estudo de caso múltiplo que fatores como falta de apoio da direção, falta de capacitação da área de Segurança da Informação, influência da cultura local, falhas na análise de riscos e resistência à mudança foram identificados como obstáculos.*

Abstract. *The adoption of the ISO 27001 standard grows among organizations worldwide. However, many challenges are faced by companies to implement this standard. There are few studies on the challenges facing Brazilian companies to establish and maintain the Information Security Management System. This article aims to identify and analyze the challenges faced in establishing and maintaining this management system on the national scene. Through the multiple case study method that factors such as lack of management support, lack of training in the Information Security area, influence of local culture, failures in risk analysis and resistance to change were identified as obstacles.*

1. Introdução

As informações desempenham papéis estratégicos fundamentais dentro das organizações, dessa forma, elas acabam sendo cobiçadas tornando-se alvo de ataques que buscam infringir sua confidencialidade, integridade e disponibilidade. As informações precisam ser protegidas para ajudar a garantir o capital das organizações.

Especialistas como Solms (1999) acreditam que o estabelecimento de normas internacionais de segurança da informação é um ponto de partida essencial para melhorar a segurança da informação de uma organização.

Para garantir esta proteção de forma eficaz, existe um sistema de gestão específico que oferece uma estrutura de controles que pode ser aplicada pelas empresas de diferentes ramos de atuação, denominado Sistema de Gestão de Segurança da Informação (SGSI).

Este sistema de gestão provê um modelo internacionalmente comprovado para, segundo a ISO 27001 (2005), estabelecer, operar, monitorar e analisar criticamente ambientes organizacionais sob o aspecto de segurança da informação.

Como ferramenta utilizada para aplicar controles de segurança da informação e obter o nível seguro de proteção existe a norma internacional de segurança da informação ISO 27001. É uma norma escrita pelos melhores especialistas de todo o mundo em segurança da informação. Sua finalidade é fornecer uma metodologia para estabelecer a segurança da informação em uma organização [Kosutic 2013].

A ISO 27001 tem como abordagem a gestão de riscos para alcançar a segurança da informação eficaz através do uso contínuo de métodos de risco, incorporadas ao modelo de processo PDCA, para monitorar, manter e melhorar a eficácia dos controles de segurança [ISO 27001 2005].

Desafios para estabelecimento e manutenção da ISO 27001 foram identificados em âmbito global. A publicação feita pela *The British Assessement Bureau* (2013) cita como desafios: o medo ou constrangimento de não conseguir a certificação depois de ser auditado, os custos iniciais e de manutenção que a certificação exige e também o fato de ser mais fácil reivindicar o cumprimento da norma do que realmente demonstrar como cumpri-la.

O estudo supracitado realizado pelo *The British Assement Bureau* (2013) possui abrangência mundial. É escassa a quantidade de estudos no cenário nacional que abordam as dificuldades enfrentadas pelas empresas brasileiras para estabelecer e manter um Sistema de Gestão de Segurança da Informação. Com base no levantamento bibliográfico de pesquisas sobre este tema, houve forte dificuldade em buscar estudos de empresas brasileiras de ramos diferentes de atuação, foram encontrados estudos de caso único sobre implementação da norma ISO 27001. Estudos de caso múltiplos foram possíveis de localizar somente em empresas estrangeiras. Por mais que os estudos destas empresas contribuam para identificar os desafios, é importante obter uma visão holística para perceber a realidade enfrentada pelas empresas brasileiras ao estabelecer e manter um Sistema de Gestão de Segurança da Informação.

Sendo assim, este trabalho procura responder a questão de pesquisa: quais são os principais desafios, no cenário nacional, a fim de estabelecer e manter um Sistema de Gestão de Segurança da Informação?

Para responder esta questão de pesquisa, o seguinte objetivo geral foi definido: identificar e analisar os principais desafios ao estabelecer e manter um Sistema de Gestão de Segurança da Informação através de um número limitado de empresas brasileiras que representam os principais ramos de atuação que mais possuem certificação na norma ISO 27001. Para atingir este objetivo geral, os objetivos específicos estabelecidos foram: desenvolver um instrumento de coleta de dados adequado ao propósito do trabalho e organizar o descrever os dados coletados

Para atingir os objetivos propostos e, consequentemente, obter a resposta da questão de pesquisa, este artigo foi estruturado da seguinte forma: a seção 2 relaciona as pesquisas que já foram feitas sobre este mesmo tema; a seção 3 descreve a metodologia que foi aplicada nesta pesquisa e suas características; a seção 4 descreve os resultados obtidos interpretados da análise dos dados coletados nas entrevistas com as empresas. Por fim, na seção 5 encontra-se a conclusão da pesquisa e os trabalhos futuros que poderão ser iniciados com base nos resultados deste trabalho.

2. Trabalhos Relacionados

Os trabalhos pesquisados sobre o tema deste artigo foram organizados conforme representa a Tabela 1.

Tabela 1. Trabalhos relacionados

Autor	Escopo	Dificuldades Identificadas
Singh et al. (2012)	Organizações da Índia.	Falta de avaliação precisa dos ativos das empresas; baixo comprometimento da direção; resitência à mudança; falta de experiência da equipe; não entendimento claro da norma ISO 27001.
Waluyan et al. (2010)	Multinacionais no Brasil.	Diferenças culturais entre os colaboradores; dificuldade em gerenciar informações confidenciais; baixa flexibilidade da norma ISO 27001.
Martins e Santos (2005)	Estudo de caso único de uma empresa brasileira.	Falta de conhecimento na área de segurança da informação; falta de *budget;* falta de interesse da direção.
Al-Awadi e Renaud (2008)	Organizações governamentais em Omã, na Arábia.	Falta de treinamento dos colaboradores; falta de entendimento dos valores de segurança por parte da área de TI; problemas de *budget* ; falta de adaptação dos colaboradores aos requisitos da norma.
Abusaad et al. (2011)	Organizações na Arábia Saudita.	Dificuldade em identificar corretamente os ativos das organizações; falta de experiência das equipes para implementação dos requisitos da norma; resistência à mudança; fraco envolvimento da direção; influência da cultura local.

3. Metodologia

Nesta pesquisa, um estudo de caso múltiplo foi desenvolvido para analisar o estabelecimento e manutenção do Sistema de Gestão de Segurança da Informação de organizações brasileiras de diferentes ramos de atuação. Entrevistas de abordagem qualitativa com os responsáveis por segurança da informação foram realizadas nestas organizações, baseando-se em um roteiro específico previamente elaborado. Segundo Malhotra (2006), este é um estudo exploratório, pois possibilita desenvolver hipóteses sobre o tema que está sendo estudado. Ao final da pesquisa, hipóteses foram levantadas sobre os possíveis desafios identificados e analisados sobre as empresas selecionadas.

A amostragem das empresas foi classificada como não probabilística, ela não utiliza seleção aleatória, confia no julgamento pessoal do pesquisador. Utilizou-se a técnica de amostragem por conveniência devido às limitações de buscar uma relação de todas as empresas brasileiras certificadas na ISO 27001, ou que possuam um Sistema de Gestão de Segurança da Informação estabelecido. Esta técnica mostra-se adequada a este tipo de pesquisa, uma vez que, segundo Malhotra (2006), a seleção das unidades amostrais é deixada a cargo do entrevistador (Tabela 2).

Tabela 2. Perfis das empresas selecionadas

Ramo	Colaboradores	Tempo de Mercado	SGSI	Período
Indústria	780	12 anos	Estabelecido	07 anos
Financeiro	160	07 anos	Estabelecido	04 anos
T.I.	174	10 anos	Estabelecido	02 anos
T.I.	80	20 anos	Estabelecido	03 anos
e-commerce	1	14 anos	Estabelecido	04 anos
Segurança Informação	60	12 anos	Certificado	03 anos

Foram selecionadas empresas brasileiras sabidamente certificadas na norma ISO 27001 ou que já possuem o Sistema de Gestão de Segurança da Informação estabelecido. Para que uma empresa estabeleça este sistema de gestão, ela precisará definir um escopo, ou seja, sobre quais os processos da empresa que o Sistema de Gestão de Segurança da Informação será implementado. A empresa de Tecnologia de Informação com 80 colaboradores possui como escopo *Data Center* e os escopos das demais empresas são todos os processos de negócio, de acordo com suas respectivas áreas de atuação.

Para assegurar a relevância das empresas selecionadas como representação do cenário nacional, os ramos de atuação fazem parte do *Top Five* mundial de seguimentos que mais possuem certificação na norma ISO 27001 e também do *Top Three* de ramos de atuação de empresas brasileiras que mais possuem certificação nesta norma, segundo levantamento realizado pela ISO (2013).

3.1. Coleta dos dados

As entrevistas presenciais e remotas foram realizadas utilizando um roteiro de entrevistas como base. A ideia do roteiro foi questionar os entrevistados sobre o ambiente organizacional e sua relação com o Sistema de Gestão de Segurança da Informação.

3.1.2. Caracterísicas do roteiro

Para estruturar o roteiro, as questões abrangem todas as etapas do ciclo PDCA aplicado à norma ISO 27001 (Figura 1). Cada questão possui objetivos para avaliar se a organização está seguindo o PDCA que a norma exige, identificando os principais problemas e desafios enfrentados para estabelecer e manter o Sistema de Gestão de Segurança da Informação. As perguntas foram divididas em duas categorias: estabelecer e manter, uma vez que o ciclo PDCA da norma visa estabelecer e manter um SGSI, de um modo geral.

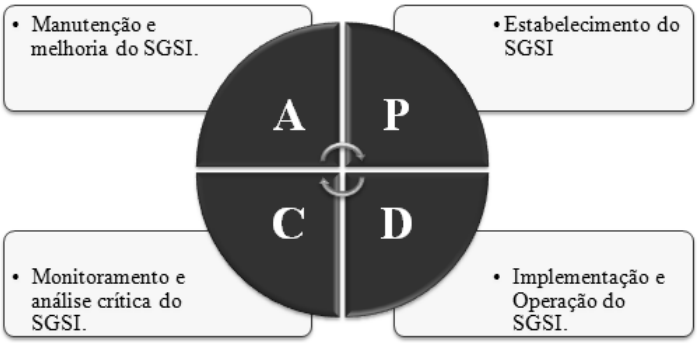


Figura 1. Ciclo PDCA aplicado ao Sistema de Gestão de Segurança da Informação

O PDCA é um modelo que busca tornar os processos da gestão de uma empresa mais ágeis, claros e objetivos. Pode ser utilizado em qualquer tipo de empresa e é divido em quatro etapas: PLAN (planejar), DO (fazer), CHECK (verificar) e ACT (agir).

O roteiro de entrevistas foi estruturado com 13 questões que são correlacionadas, com o objetivo de identificar inconsistências nas respostas coletadas dos entrevistados. Além disso, o roteiro possui outras características que foram utilizadas para sua estruturação, conforme representadas na Tabela 3:

Tabela 3. Estruturação do roteiro de entrevistas

Característica	Descrição	Referência
Abordagem	Tipo funil: perguntas genéricas progredindo para específicas.	Malhotra (2006)
Estrutura	Perguntas abertas. Objetivo de buscar os maiores detalhes possíveis das respostas dos entrevistados, devido a complexidade do tema.	Trivinos (1990)
Enunciado	Utilizadas palavras comuns conhecidas por quem atua nesta área. Não foram uilizadas palavras ambíguas, com alternativas implícitas, suposições implícitas, generalizações e estimativas.	Malhotra (2006)
Objetivos	Cada questão possui um objetivo que descreve o que de fato está sendo buscado como resposta no enunciado das perguntas.	Malhotra (2006)
Nível	Divididas nas categorias "mais vigorosas" e "menos vigorosas". As questões "mais vigorosas" provocam pensamentos mais profundos, já as "menos vigorosas" constituem-se em respostas mais objetivas	Siqueira (2011)

Depois de estruturado, o roteiro foi previamente avaliado e aprovado por especialistas em segurança da informação que possuem certificações como, por exemplo, CISSP, CISM, auditor líder em ISO 27001, entre outras capacitações.

3.2. Análise dos dados

A técnica de Análise de Conteúdo foi utilizada para a análise de dados desta pesquisa. Segundo Moraes (199), esta técnica mostra-se mais adequada para descrição e interpretação de conteúdos de qualquer classe de documentos. Esta técnica permite uma melhor compreensão dos significados dos textos.

A técnica de Análise de Conteúdo dos dados foi dividida em cinco etapas, baseando-se nas sugestões de Moraes (1999): preparação, onde os dados foram transcritos para preparação; unitarização, onde foram identificadas as unidades de registro; categorização, onde os dados resultantes das unidades de registros foram separados de acordo com os termos comuns; descrição, onde um texto síntese por categoria foi elaborado de acordo com as respostas dos entrevistados e, por fim, interpretação, quando os dados descritos nas categorias foram interpretados.

4. Resultados Obtidos

As hipóteses identificadas resultantes da técnica de análise de foram: Falta de apoio da alta direção, Falta de capacitação da equipe de Segurança da Informação, Influência da cultura local, Falhas na elaboração da Análise de Risco e Resistência à mudança.

4.1. Falta de apoio da alta direção

O comprometimento da direção e todos os níveis gerenciais é primordial no estabelecimento e manutenção do Sistema de Gestão de Segurança da Informação, conforme mencionado na ISO 27001.

Fatores que caracterizam esta falta de comprometimento foram identificados nas respostas coletadas como: não provimento de recursos para realização de programas que visam expandir a cultura de segurança da informação dentro das organizações, pouco envolvimento nas ações de segurança da informação com o intuito de demonstrar aos colaboradores que a segurança da informação é uma preocupação oriunda do negócio da organização, falta de análises críticas do sistema de gestão para assegurar a melhoria

contínua nos processos e alinhamento dos objetivos da empresa para, não somente permanecer em conformidade com a norma ISO 27001, mas também garantir que todos os processos estejam alinhados e com os mesmos objetivos dentro da organização.

A partir deste desafio, outros poderão ser reduzidos consideravelmente. Um exemplo disso é o provimento de recursos para capacitação das equipes de Segurança da Informação. Uma direção fortemente comprometida com a segurança dispõe de recursos para que sua equipe esteja sempre capacitada a orientar seus colaboradores e utilizar-se das melhores práticas no mercado, incluindo a aplicação dos controles da norma ISO 27001 de forma mais consistente e de acordo com a realidade da organização.

4.2. Falta de capacitação da equipe de SI (Segurança da Informação)

A identificação deste desafio partiu não somente das respostas explícitas dos entrevistados, mas, também, das respostas implícitas.

Alguns entrevistados demonstraram sólidos conhecimentos da área de segurança da informação de suas empresas, porém, determinados problemas estavam sendo causados pela própria área de segurança da informação, não por má fé da equipe, mas puramente pela falta de capacitação e experiência.

Exemplos disso são a não necessidade de medição de determinados controles de segurança da informação e orientações específicas para especialistas de Tecnologia da Informação. Dentre as respostas coletadas, houve casos em que a área de segurança da informação sequer sabia responder o que de benefício para a organização o Sistema de Gestão de Segurança da Informação trouxera.

Além disso, existem áreas de segurança da informação que não possuem um entendimento completo da norma ISO 27001. Elas possuem uma visão deturpada do que é de fato um Sistema de Gestão de Segurança da Informação estabelecido. Um exemplo é a aplicação somente dos controles de segurança da informação no ambiente da empresa, sendo que, para que este sistema de gestão seja adequadamente estabelecido deve ter as etapas correspondentes ao ciclo do PDCA implantadas, executadas, medidas e melhoradas.

Entrevistados apontaram, também, que a mão-de-obra não capacitada estava impactando no processo do sistema de gestão como um todo, de tal forma que os incidentes de segurança da informação não estavam sendo solucionados devido a este despreparo.

4.3. Influência da cultura local

Este desafio acaba aparecendo de forma onipresente entre as respostas dos entrevistados. A influência da cultura local, segundo as respostas obtidas, acaba aparecendo como fator que origina outros desafios como, por exemplo, a falta de comprometimento dos colaboradores para com a cultura de segurança da informação das empresas.

É de senso comum que a cultura local do Brasil referente à segurança da informação precisa evoluir. Segundo informações obtidas dos entrevistados, grande parte dos usuários ainda tem a ideia de que segurança da informação é somente “proteger o computador” e, dessa forma, acabam não valorizando as informações confidenciais que são trocadas por outros meios como, por exemplo, informações faladas em locais inadequados, materiais com informações confidenciais descartados de forma incorreta.

Colaboradores atribuindo acessos confidenciais sem um estudo prévio do que realmente é necessário atribuir de acessos, compartilhamento de senhas pessoais em

situações de ausência de colaboradores ou para divisão de atividades, falta de apoio dos gestores das áreas de negócio na expansão da cultura de segurança para seus subordinados, excesso de confiança nos colegas de trabalho fazendo com que informações confidenciais sejam expostas em locais indevidos. Ou seja, a cultura de que as situações devem ser tratadas e resolvidas de forma rápida, fazendo com que a segurança fique em segundo plano.

Um fato interessante observado nas respostas dos entrevistados aponta para a falta de interesse dos colaboradores em abrir incidentes de segurança da informação. Alguns entrevistados acabaram relatando que muitos colaboradores ainda têm o pensamento de que a abertura de incidentes é somente tarefa da área de Segurança da Informação.

4.4. Falhas na elaboração da análise de risco

As falhas na elaboração da análise de risco desencadeiam outros desafios, assim como a falta de apoio da alta direção. Uma análise de riscos mal feita é um dano estrutural no Sistema de Gestão de Segurança da Informação, pois é a base do processo como um todo. É da análise de riscos que os ativos do escopo deste sistema de gestão são identificados e, a partir destes ativos, as políticas de segurança da informação e toda uma cadeia de processos serão elaboradas.

Segundo os dados coletados nas entrevistas, a ineficiência em identificar os ativos das organizações para definição dos escopos que serão abrangidos pelo Sistema de Gestão de Segurança da Informação acaba fazendo com que a análise de risco não cubra todas as arestas necessárias. Além disso, os fatores motivadores para estabelecimento deste sistema de gestão também acabam impactando a elaboração da análise de riscos.

Alguns entrevistados relataram que a análise de riscos já estava definida de acordo com outros padrões de segurança mais técnicos, diferentes da norma ISO 27001, e que a partir desta análise de riscos, o Sistema de Gestão de Segurança da Informação foi estabelecido. Exemplo disso é uma análise de riscos feita para atender aos requisitos da norma internacional PCI-DSS (utilizado em empresas com grande volume de transações de cartão de crédito). Os requisitos para a análise de riscos desta norma, por mais que também estejam ligados fortemente à segurança da informação, não atendem a determinados requisitos da norma ISO 27001 e, mesmo assim, foram utilizados como base para estabelecer o Sistema de Gestão de Segurança da Informação.

4.5. Resistência à mudança

Qualquer norma de gestão enfrenta este desafio antes mesmo da norma ser estabelecida no ambiente. O fato de grande parte dos colaboradores ainda terem o pensamento de que segurança da informação é responsabilidade somente de uma área específica, acaba fazendo com que os mesmos resistam a seguir as políticas de segurança da informação e as boas práticas divulgadas pela empresa.

Boa parte das atividades e controles gerados pelas políticas de segurança da informação, por exemplo, são vistos como um "atraso" nos processos de negócio, segundo relatos dos entrevistados. Muitas dessas ideias deturpadas em relação à segurança da informação são fomentadas pelo não conhecimento ou não valorização que as informações exercem sobre o negócio como um todo.

Implantação de novas tecnologias, a inclusão de mais controles de segurança, em geral tudo que gera mais esforço por parte dos colaboradores acaba sendo encarado como

atividade burocrática, sem resultados mensuráveis. Cabe aí, portanto, reiniciando o ciclo dos desafios identificados nesta pesquisa, um maior apoio da direção para proporcionar subsídios humanos e técnicos para demonstrar no que, de fato, esses "esforços extras" dos colaboradores estão contribuindo para o ambiente organizacional da empresa para assim, quem sabe, a resistência à mudança acabe dando lugar à conscientização à segurança da informação.

4.6. Outras constatações

Durante a fase de análise de dados, foi possível identificar outras constatações importantes que este trabalho contribuiu, como: os desafios enfrentados por cada etapa do ciclo PDCA, fatores motivadores para o estabelecimento do Sistema de Gestão de Segurança da Informação e os principais benefícios identificados pelas empresas pesquisadas.

Apesar do objetivo deste trabalho ser identificar os desafios de forma geral para estabelecimento e manutenção do Sistema de Gestão de Segurança da Informação, este trabalho também acabou contribuindo para apresentar os obstáculos relacionados a cada etapa do PDCA (Tabela 4).

Tabela 4. Desafios enfrentados por cada etapa do ciclo PDCA

Desafios	Etapas
Falta de apoio da alta direção	Plan, Do, Check, Act
Falta de capacitação da equipe de Segurança da Informação	Plan, Do, Check, Act
Influência da cultura local	Do, Act
Falhas na elaboração da Análise de Risco	Plan
Resistência à mudança	Do, Act

Alguns dos principais fatores motivadores identificados nas respostas foram: exigência por parte da matriz, vantagem competitiva de mercado, busca por um ambiente processual padronizado e controlado, almejar um ambiente seguro culminando em uma certificação na ISO 27001 e proteção das informações confidenciais das organizações.

Além disso, foi possível identificar os principais benefícios que o estabelecimento deste sistema de gestão está trazendo para as organizações: melhorias de imagem e marketing das empresas, aumento da disponibilidade dos ambientes de infraestrutura de Tecnologia da Informação, diminuição nos custos com infraestrutura de Tecnologia da Informação, apoio importante no processo de Governança de TI, mapeamento das falhas de segurança dos ambientes organizacionais e credibilidade perante aos clientes.

Ao final desta pesquisa, também foi possível fazer uma comparação dos resultados obtidos com os desafios identificados pelos trabalhos relacionados, onde ficou evidente a semelhança dos resultados do cenário nacional com os estudos realizados na Índia, Omã e Arábia Saudita.

5. Conclusão

O presente artigo buscou responder a questão de pequisa: quais são os principais desafios, no cenário nacional, a fim de estabelecer e manter um Sistema de Gestão de Segurança da Informação? Esta questão foi respondida com sucesso, respeitando a amostra selecionada que representa o cenário brasileiro.

Os dados coletados nas entrevistas foram analisados chegando-se a identificação destes desafios através de cinco hipóteses representadas por categorias. São elas: Falta de

apoio da alta direção, Falta de capacitação da equipe de segurança da informação, Influência da cultura local, Falhas na elaboração da análise de risco e Resistência à mudança. Cada uma destas categorias descreve a síntese dos problemas citados pelos entrevistados do Sistema de Gestão de Segurança da Informação da organização.

Dificuldades tiveram que ser superadas ao longo desta pesquisa para obtenção dos objetivos propostos. Exaustivos testes na elaboração e aprovação do roteiro de entrevista, dificuldades em conseguir flexibilidade das empresas selecionadas para realização das entrevistas, as frustrações momentâneas enfrentadas nos cancelamentos de entrevistas por motivos diversos, a seleção de outras empresas que atendessem aos requisitos desta pesquisa e, por fim, a própria análise de dados que foi realizada com a paciência e os cuidados que esta fase requer.

Como resultado de uma análise de dados criteriosa, foi possível obter outras constatações que não estavam entre os objetivos desta pesquisa. Além de identificar os desafios que impedem a adesão em massa de empresas brasileiras à norma ISO 27001 de forma geral, esta pesquisa contribuiu para identificar estes obstáculos através de cada etapa do ciclo PDCA, os principais fatores motivadores e os principais benefícios que estas empresas brasileiras estão obtendo com o estabelecimento do Sistema de Gestão de Segurança da Informação.

Na comparação dos resultados desta pesquisa com os trabalhos relacionados, percebe-se que os desafios identificados neste artigo assemelham-se consideravelmente com os desafios dos estudos realizados na Índia, Omã e Arábia Saudita. Estes dados são interessantes, pois existe uma diferença cultural forte entre o Brasil e os países mencionados e, mesmo assim, os desafios acabaram convergindo-se.

Sendo assim, os resultados obtidos nesta pesquisa reforçam a ideia de que esse artigo possa ser utilizado como um guia para contribuir de forma preventiva, antecipando aos especialistas em Segurança da Informação, os principais desafios que poderão ser enfrentados para o estabelecimento e manutenção do Sistema de Gestão de Segurança da Informação.

5.1. Trabalhos futuros

Análises aprofundadas sobre as causas dos desafios mencionados neste artigo poderão ser realizadas. Com as causas mapeadas, medidas preventivas poderão ser elaboradas e aplicadas para evitar ou minimizar a ocorrência dos desafios citados.

Como consequência desta pesquisa, também poderá ser conduzido um estudo focado em sugerir e aplicar possíveis soluções aos desafios detectados neste trabalho.

Adaptação do roteiro de entrevistas para utilização em pesquisas que tenham como foco outros escopos, ramos de atuação específicos ou determinadas regiões geográficas.

E, por fim, realizar um estudo mais específico que possa levantar hipóteses para explicar os motivos dos desafios citados neste artigo assemelharem-se com os estudos realizados em outras regiões geográficas com culturas diferentes.

Referências

ABNT NBR ISO/IEC 27001:2005, (2005) “Tecnologia da informação — Técnicas de segurança — Sistemas de gestão de segurança da informação — Requisitos”.

Abusaad, Belal, Saeed Fahad A., Alghathbar, Khaled, Bilal, Khan. (2011) "Implementation of ISO 27001 in Saudi Arabia – Obstacles, Motivations, Outcomes and Lessons Learned", 9th Australian Information Security Management Conference, Edith Cowan University. December.

Al-Awadi, Maryam, Renaud Karen. (2008) "Success Factors in Information Security Implementation in Organizations", University of Glasgow.

British Assessment Bureau, (2013) "Survey Shows Fear of ISO 27001", http://www.british-assessment.co.uk/news/survey-shows-fear-of-iso-27001, Julho.

ISO, International Organization for Standardization, (2013), "ISO Survey 2012", http://www.iso.org/iso/home/standards/certification/iso-survey.htm?certificate=ISO%209001&countrycode=AF, Setembro.

Kosutic, Dejan (2013) "We have implemented ISO 9001, can something be used for ISO 27001 / ISO 22301 / BS 25999-2?" IS&BCA. http://support.epps.eu/customer/portal/articles/787939-we-have-implemented-iso-9001-can- something-be-used-for-iso-27001-iso-22301-bs-25999-2-, Outubro.

Malhotra, N. K. (2006) "Pesquisa de marketing: uma orientação aplicada", Porto Alegre: Bookman.

Moraes, Roque (1999) "Análise de conteúdo", Revista Educação, Porto Alegre, Vol. 22, Nº. 37, pp. 7-32.

Martins, Alaíde, Santos, Celso (2005) "Uma Metodologia para Implantação de um Sistema de Gestão de Segurança da Informação", Journal of Information Systems and Technology Management, vol 2, Nº 2, pp. 121-136. Salvador.

Singh, Abhay, Sharma, Sammarth, Pandey, Manish, Chaurasia, Sandarbh, Vaish, Anaurika, Venkatesan S. (2012) "Implementation of ISO 27001 in Indian Scenario: Key Challenges", International Conference on Recent Trends of Computer Technology in Academia.

Siqueira, Jairo, (2011) "A Arte das Perguntas Criativas e Desafiadoras", http:// http://criatividadeaplicada.com/2011/07/28/a-arte-das-perguntas-criativas-edesafiadoras/, Julho.

Solms, Von R. (1999) "Information Security Management: Why Standards are Important", Information Management & Computer Security. vol. 46, n° 8, p. 91-95.

The British Assessment Bureau, (2013) "Key Survey Illustrates the Importance of ISO 27001", http://www.british-assessment.co.uk/news/key-survey-illustrates-the-importance-of-iso-27001, Agosto.

The British Assessment Bureau, (2013) "Survey Shows Fear of ISO 27001", http://www.british-assessment.co.uk/news/survey-shows-fear-of-iso-27001, Julho.

The Trivinos, Augusto Nibaldo Silva (1990) "Introdução à Pesquisa em Ciências Sociais: A Pesquisa Qualitativa em Educação", São Paulo, Atlas, 1990. p. 146.

Waluyan, Liska, Blos, Mauricio, Nogueira, Stephanie, Asai, Tatuso. (2010) "Potential Problems in People Management concerning Information Security in Cross-cultural Environment – The Case of Brazil", Journal of Information Processing, Vol. 18, pp. 38-42. February.

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

WGID – IV Workshop de Gestão de Identidades Digitais

Painel:

Gestão de Identidade Eletrônica e Identificação Civil no Brasil

Panelistas:

Hélvio Pereira Peixoto[1], **Rafael Timóteo de Sousa Júnior**[2], **José Alberto Torres**[3]

Moderadora:

Michelle Silva Wangham[4]

[1]Comitê Gestor do Sistema Nacional de Registro de Identificação Civil
Ministério da Justiça do Brasil

[2]Departamento de Engenharia Elétrica
Universidade de Brasília

[3]Infraestrutura Tecnológica do Registro de Identificação Civil
Ministério da Justiça do Brasil

[4]Grupo de Sistemas Embarcados e Distribuídos
Universidade do Vale do Itajaí

Resumo. *O Registro de Identificação Civil (RIC) é um projeto do Ministério da Justiça concebido para criar uma nova cédula de identidade. Esta cédula deverá ter um chip (tecnologia smart card) para maior segurança e flexibilidade, sendo similar a um cartão bancário. Ela reunirá todos os dados da cédula de identidade atual, bem como CPF e número de título de eleitor, dentre outras informações. O RIC será integrado com o sistema de identificação de impressões digitais (AFIS) e deverá integrar todos os bancos de dados de identificação do Brasil. Os principais objetivos do projeto RIC são: (i) criar um número único de Registro de Identidade Civil - RIC - no contexto do Sistema Nacional de Registro de Identificação Civil (SINRIC), (ii) criar um órgão central coordenado com os órgãos estaduais de identificação, para a emissão, em âmbito nacional, da nova cédula de Identificação Civil com recursos modernos de segurança e de certificação digital. Este projeto foi oficialmente lançado em 2010 e, após uma fase inicial de testes em alguns municípios brasileiros, caminha para uma nova fase em que várias questões verificadas nos testes serão devidamente tratadas. Este painel tem por objetivo apresentar e discutir com a comunidade científica os avanços obtidos com o RIC até o momento e os novos desafios que se apresentam para as fases futuras.*

Apresentação do Comitê Técnico de Gestão de Identidades da Rede Nacional de Ensino e Pesquisa (RNP)

Coordenador:

Marco Aurélio Amaral Henriques[1]

[1]Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (Unicamp)
maah@unicamp.br

Resumo. *O Comitê Técnico de Gestão de Identidades tem como objetivo realizar recomendações técnicas para apoiar as atividades de Governança da RNP, em particular aquelas do Comitê Assessor de Gestão de Identidade. A composição deste comitê técnico é de especialistas que atuam na prospecção de temas relacionados a gestão de identidades. Dentre as atividades de prospecção que se tornaram serviços regulares na RNP podem ser citadas a Infraestrutura de Chaves Públicas de Ensino e Pesquisa (ICPEdu), que permite a emissão, ampla utilização e o reconhecimento/validação de certificados digitais entre instituições de ensino e pesquisa; a Comunidade Acadêmica Federada (CAFe), que viabiliza o acesso de pesquisadores de instituições brasileiras a uma série de informações nacionais e internacionais mediante o uso de um único par usuário/senha; a infraestrutura de rede sem fio de escopo mundial eduroam, que permite a um pesquisador de uma instituição participante usar a infraestrutura de rede sem fio (WiFi) de outra instituição sem necessidade de cadastro prévio, já que suas credenciais na instituição de origem são reconhecidas por todas as demais; dentre outros. Além de seus membros convidados, a participação nas atividades é aberta para outros interessados que possam oferecer expertise, recursos humanos, equipamentos e/ou serviços apropriados ao desenvolvimento de suas atividades. O objetivo desta apresentação é divulgar as atividades em curso no Comitê Técnico de Gestão de Identidades e convidar novos membros da comunidade a se juntarem a ele.*

GIdLab: Laboratório de Experimentação em Gestão de Identidade*

Maykon Chagas de Souza[1], **Emerson Ribeiro de Mello**[2], **Michelle Silva Wangham**[1]

[1]Universidade do Vale do Itajaí – (UNIVALI)

[2]Instituto Federal de Santa Catarina – (IFSC)

{mchagas,wangham}@univali.br, mello@ifsc.edu.br

1. Projeto GIdLab

A Rede Nacional de Ensino e Pesquisa (RNP) oferece dois serviços na área de Gestão de Identidades: a Infraestrutura de Chaves Públicas para ensino e pesquisa (**ICPedu**[1]) e a Comunidade Acadêmica Federada (**CAFe**[2]), baseada no *framework* Shibboleth[3]. Estes serviços não permitem em suas políticas de uso que pesquisadores os utilizem para realizar seus experimentos. Constatou-se que os pesquisadores dedicavam uma quantidade razoável de tempo para construir seu próprio ambiente, executar seus experimentos e depois se desfazer do ambiente, uma vez que é custoso manter disponível o ambiente. Configurar uma federação ou uma infraestrutura de chaves públicas (ICP) para realizar experimentos pode ser uma tarefa mais árdua e demorada do que a implementação da pesquisa propriamente dita.

O projeto GIdLab foi criado em fevereiro de 2013 por uma iniciativa do Comitê Técnico de Gestão de Identidades (CT-GID) da RNP e, desde então, é mantido pela RNP como plataforma de apoio aos pesquisadores brasileiros na área de gestão de identidades. O projeto objetiva estimular e facilitar o desenvolvimento de novas soluções que possam vir a ser disponibilizadas como serviços pela RNP.

Esta palestra visa apresentar o **GIdLab**[4], um laboratório para realização de experimentos em gestão de identidades (GId), que tem por objetivo disponibilizar aos pesquisadores um ambiente de testes para que estes possam conduzir experimentos com diferentes Infraestruturas de Autenticação e de Autorização (IAAs) e com a Infraestrutura de Chaves Públicas para ensino e pesquisa (ICPedu).

2. Infraestrutura do GIdLab

O GIdLab disponibiliza aos pesquisadores os seguintes serviços:

- Uma federação Shibboleth, chamada **CAFe Expresso**, com provedores de identidade (IdPs) e provedores de serviço (SPs), além de Serviços de Descobertas (DS) e o serviço *uApprove*;
- Uma federação SAML baseada no *framework* SimpleSAMLphp;
- Um provedor OpenID Connect;

*Projeto financiado pela RNP.

[1]http://www.rnp.br/servicos/servicos-avancados/icpedu

[2]http://www.rnp.br/servicos/servicos-avancados/cafe

[3]http://shibboleth.net

[4]http://wiki.rnp.br/display/gidlab

- Sistema de Gerenciamento de Certificados Digitais (SGCI) da Infraestrutura de Chaves Públicas para Ensino e Pesquisa (ICPedu) configurado para realização de experimentos;
- Autoridade Certificadora Online (AC-Online);
- Repositório com máquinas virtuais configuradas para implantar uma federação Shibboleth (IdP, SP e WAYF);

Em 2014, a infraestrutura do GIdLab conta com trinta e cinco (35) Máquinas Virtuais (VMs) distribuídas nos Pontos de Presença (PoPs) da RNP, sendo algumas destas dedicadas para projetos específicos que fazem uso do GIdLab.

A Figura 1 ilustra os serviços oferecidos e as infraestruturas de autenticação e de autorização utilizadas no GIdLab. Em novembro de 2014, o GIdLab disponibilizará uma federação tendo como base o *framework OpenAM*.

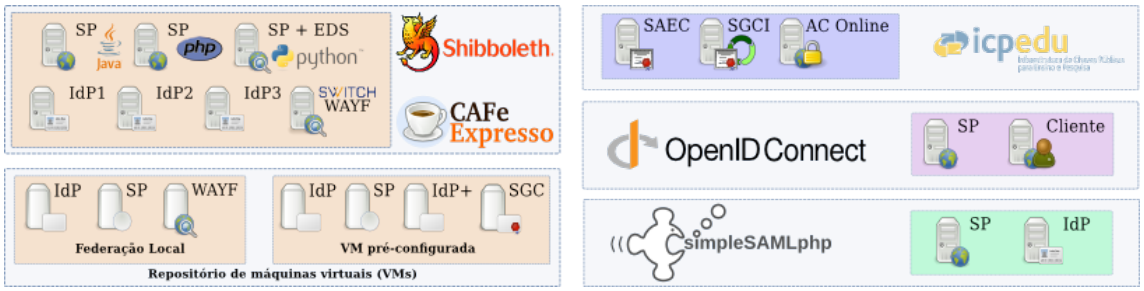


Figura 1. Infraestrutura disponível no GIdLab

O GIdLab foi disponibilizado em Julho de 2013. No ano de 2013, o projeto atendeu 4 projetos que utilizaram dos recursos da CAFe Expresso, OpenID Connect e também do ICPEdu. Em 2014, nove projetos fizeram o cadastro no GIdLab, sendo que destes 5 ainda estão em andamento.

3. Público alvo

O GIdLab tem como público alvo pesquisadores interessados em desenvolver experimentos práticos com Infraestruturas de Autenticação e Autorização e com a ICPEdu, em especial, os pesquisadores participantes do Programa de Gestão de Identidades (PGId) e dos Grupos de Trabalhos (GTs) da RNP.

4. Como Utilizar o GIdLab

Para conduzir experimentos no GIdLab, o pesquisador precisa preencher um formulário para cadastro de seu projeto de experimentos no site do Projeto GIDLab[5]. Após o cadastro, a equipe do GIdLab entrará em contato com os pesquisadores reportando o recebimento do cadastro e criará contas de acesso e, quando necessário, auxiliará a execução dos experimentos. Os pesquisadores terão acesso aos serviços e, quando necessário, terão acesso as máquinas virtuais do GIdLab.

[5]http://bit.ly/formularioCadastroGIdLab

WGID - Artigos Completos

Controle de Acesso Baseado em Políticas e Atributos para Federações de Recursos

Edelberto F. Silva[1], **Débora Muchaluat-Saade**[1] **e Natalia C. Fernandes**[1]

[1]Universidade Federal Fluminense (UFF) – Laboratório MídiaCom – Niterói, RJ – Brasil

Abstract. *Federated authentication methods as a base for accessing resources in virtual organizations is a challenge for the academic community and, therefore, the aim of this work. Issues about maintaining user attributes as well as the use of different access policies must be modeled and evaluated to allow a better management of identities in this environment. This work proposes an architecture for policy-based access control and also implements and validates an attribute provider using the CAFe academic federation (CAFeExpresso) to access the resources of academic virtual organizations.*

Resumo. *A introdução de métodos de autenticação federada como base para o acesso aos recursos de organizações virtuais é de grande interesse para a comunidade acadêmica e, por essa razão, alvo deste trabalho. Questões relacionadas ao armazenamento de atributos de usuários, assim como a utilização de diferentes políticas de acesso devem ser modeladas e avaliadas para permitir uma melhor gestão de identidade nesse ambiente. Este trabalho propõe uma arquitetura para controle de acesso baseado em políticas e implementa e valida a utilização de um provedor de atributos baseado na federação acadêmica CAFe (CAFeExpresso) para o acesso a recursos de organizações virtuais acadêmicas.*

1. Introdução

Esta pesquisa tem como principal motivação o emprego e a facilitação do ingresso das federações acadêmicas de identidades em ambientes de recursos distribuídos. Sabe-se que diversos são os esforços nesta área, principalmente quando do surgimento da computação em grade (*grid*) [Foster et al. 2001] e das redes federadas para experimentação em Internet do Futuro [Silva et al. 2013b]. Nesses cenários, os usuários advindos de diferentes instituições acessam os recursos compartilhados para desenvolver pesquisas.

Com a evolução do conceito de facilitação de ingresso dos usuários/experimentadores acadêmicos em ambientes de recursos compartilhados e distribuídos, surge então a proposta da criação de federações acadêmicas para auxiliar essa interação. Um exemplo claro de federação acadêmica criada no Brasil e em amplo crescimento e difusão é a CAFe[1] (Comunidade Acadêmica Federada). Sua principal motivação é criar uma federação de A&A (Autenticação e Autorização), ou seja, um ambiente em que as entidades envolvidas (Provedores de Serviço e Provedores de Identidade) participantes confiem uns nos outros, utilizando uma base federada de identidade para facilitar o ingresso aos serviços oferecidos por cada uma das instituições participantes. É a partir desse conceito que este trabalho se motiva, focando na união de federações (acadêmica e de recursos), porém aplicando conceitos e funcionalidades da Gestão de Identidade, GId (*Identity Management* - IdM) [Silva et al. 2013b]. A GId pode ser entendida como conjunto de processos e tecnologias usados para garantir a identidade

[1]http://portal.rnp.br/web/servicos/cafe

de uma entidade, garantir a qualidade das informações de identidade (identificadores, credenciais e atributos) e utilizar essas garantias em procedimentos de autenticação, autorização e auditoria [Silva et al. 2013b].

Desta forma, pode-se visualizar um cenário genérico de integração da federação acadêmica CAFe com qualquer federação de recursos conforme a Figura 1. Neste ambiente, há a união das federações com destaque para a autenticação e a autorização, tópicos que este trabalho objetiva tratar e propor soluções.

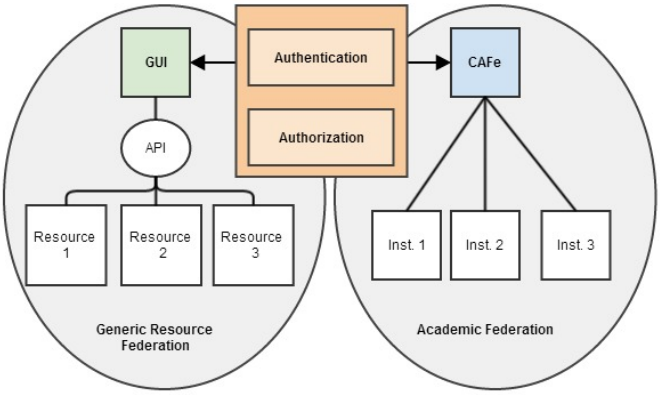


Figura 1. União das federações. Federação CAFe e federação de recursos genérica.

Neste trabalho, será proposta uma arquitetura para controle de acesso baseado em políticas e atributos para organizações virtuais acadêmicas. Como estudo de caso, uma organização virtual considerada é uma rede de experimentação de Internet do Futuro. Espera-se, em um futuro próximo, que haja maturidade para propor a adoção da proposta de controle de acesso baseado em políticas pelo projeto FIBRE (*Future Internet Testbeds Experimentation Between Brazil and Europe*)[2]. O controle de acesso baseado em políticas proposto é baseado em um provedor de atributos, que armazena atributos complementares à federação acadêmica CAFe para um dado usuário. Esses atributos complementares são aqueles empregados apenas em um contexto específico, como o de um projeto de experimentação em redes. Nesse ambiente, é necessário pensar em soluções para, por exemplo, realizar o vínculo entre o identificador do usuário na federação acadêmica e no provedor de atributos, a fim de preservar a privacidade do usuário. Este artigo detalha a solução proposta para o provedor de atributos, que foi testada e validada utilizando como ambiente de experimentação o GidLab (Laboratório de Gestão de Identidade) da RNP.

Após esta introdução tem-se a Seção 2 com os principais tipos de controle de acesso, assim como a linguagem de definição de políticas XACML *eXtensible Access Control Markup Language* [Moses 2005]. Logo após é apresentada a proposta de controle de acesso baseado em políticas e atributos para federações de recursos, na Seção 3. Já na Seção 4 são expostos os resultados para o provedor e agregador de atributos desenvolvido neste trabalho. Finalizando com a Seção 5 com as considerações finais e trabalhos futuros.

2. Controle de Acesso

Serão apresentados os modelos de controle de acesso RBAC (*Role-based Access Control*) [Ferraiolo et al. 2001], ABAC (*Attribute-based Access Control*) [Hu et al. 2014] e

[2]http://www.fibre-ict.eu/

PBAC (PBAC - *Policy-based Access Control*) [Yavatkar et al. 2000], além da linguagem de políticas e trocas de mensagens baseada em XML, o XACML (*eXtensible Access Control Markup Language*) [Moses 2005].

2.0.1. RBAC

Após a larga utilização dos modelos de controle de acesso surge o RBAC (*Role-based Access Control*). O RBAC foi bastante utilizado, em princípio, no sistema operacional UNIX, com suas bases de dados para controle de acesso de grupos [Ferraiolo et al. 2001]. Na utilização do RBAC hierárquico, os papéis (*roles*), associam-se a um conjunto de operações que podem ser realizadas por um usuário ou grupo a um ou mais objetos (*objects*). Modelos de RBAC também podem suportar hierarquia, onde o usuário (*subject*) de nível inferior de uma árvore herda parte dos direitos que usuários de níveis diretamente superiores a este (pais e filhos na árvore) possuem.

O modelo RBAC provê uma flexibilidade maior com relação a regras e aplicação em grupos. Porém, não se mostra tão flexível do ponto de vista do administrador, já que regras geralmente estão associadas diretamente a papéis. Sendo assim, quando um usuário necessita de regras específicas, diferentes daquelas associadas ao papel que ele detém, isso pode dispender bastante esforço administrativo e acrescentar dificuldade na gerência geral das regras existentes.

2.0.2. ABAC

O modelo ABAC (*Attribute-based Access Control*) [Hu et al. 2014] é, como o nome induz, um controle baseado em atributos. Ao pensar em atributos, tem-se a ideia de uma tupla *atributo=valor*. No contexto do ABAC, este conceito é empregado tanto para *subjects* (e.g. usuários), como *objects* (e.g. recursos). O ABAC não é um padrão tão bem definido como o DAC, MAC ou o RBAC, pois é resultado de diversas propostas que surgiram nos últimos anos na literatura [Jin et al. 2012]. A padronização deste modelo se deu pelo NIST (*National Institute of Standards and Technology*) [Hu et al. 2014] em 2014.

O ABAC tem como principal objetivo herdar as melhores práticas dos modelos de controle de acesso anteriores e tratar atributos mutáveis, dinâmicos e específicos de um ambiente. Em [Jin et al. 2012], é proposto um modelo ABAC flexível que possa herdar algumas características dos modelos DAC, MAC e RBAC, ou ainda, utilizar um dos modelos citados independentemente, porém, prevendo a inserção de novos atributos e permitindo um controle de acesso mais granular e escalável.

2.0.3. PBAC

O controle de acesso baseado em políticas (PBAC - *Policy-based Access Control*) [Yavatkar et al. 2000] herda muito dos conceitos aplicados à qualidade de serviço e percebe-se que pode ser aplicado em conjunto com as ideias nas quais o ABAC se apoia. Porém, o PBAC tem sido encarado mais como uma forma de implementação do modelo ABAC atualmente difundido, com a utilização de diversos atributos para a escolha de políticas específicas. O PBAC também é uma forma de se utilizar pontos de decisão e coleção de atributos específicos aos *subjects* e *resources*[3].

[3]Os *resources* são semelhantes aos *objects*, descritos no modelo anterior.

Basicamente o modelo de controle de acesso baseado em políticas apresenta quatro entidades com seus papéis bem definidos, sendo eles: *Policy Enforcement Point* (PEP): entidade do sistema que realiza o controle de acesso, realizando pedidos de decisão e executanto esses pedidos; *Policy Decision Point* (PDP): entidade que avalia a política e a torna uma decisão de autorização a ser encaminhada ao PEP; *Policy Information Point* (PIP): entidade que exerce o papel de fonte de atributos e valores para *subjects* e *resources* e *Policy Administration Point* (PAP): entidade que cria (através do administrador) uma política ou um conjunto delas[4].

2.1. XACML

A *eXtensible Access Control Markup Language* (XACML) [Moses 2005] é uma linguagem baseada em XML para a definição de políticas de segurança de forma padronizada pela OASIS (*Organization for the Advancement of Structured Information Standards*), a fim de garantir a interoperabilidade entre sistemas que desejam tratar a autorização.

Além de ser uma linguagem para políticas de controle de acesso, a XACML define também um formato para mensagens de pedido e resposta. No padrão XACML, são definidos os formatos de troca de pedidos e respostas entre as entidades PDP e PEP, onde o último efetua realmente o processamento e aplicação da política. Já para garantir a interoperabilidade entre aplicações, o XACML faz uso de uma camada de abstração entre o ambiente de aplicação e o chamado Contexto XACML. Um Contexto XACML nada mais é que a definição XML representando canonicamente as entradas e saídas do PDP.

No padrão XACML, é possível tratar políticas com mais flexibilidade utilizando um formato padrão de troca de mensagens, além de aproveitar técnicas de avaliação de políticas que auxiliam na prática do controle de acesso. Por exemplo, o XACML prevê ainda alguns algoritmos de combinação de regras (*Rule-combining algorithm*) que facilitam a aplicação das regras. Algoritmos de combinação de regras podem definir, por exemplo, que se uma das regras não for atendida, todo o acesso é negado ou vice-versa. Os algoritmos estão bem definidos na documentação disponível em [Moses 2005], assim como o formato das mensagens XACML a serem trocadas e a descrição para políticas. Vale ressaltar que o XACML prevê quatro tipos de retorno ao comparar atributos às políticas criadas. São eles: (1) *Permit*: aplicação da política de acesso permitido; (2) *Deny*: aplicação da política de acesso negado; (3) *Indeterminate*: quando um erro ocorre ou um valor requerido de um atributo não é encontrado e não é possível aplicar política alguma e (4) *Not Applicable*: a requisição não pode ser respondida (ou não se encaixa) pelo serviço.

3. Proposta de Controle de Acesso Baseado em Políticas e Atributos para Federações de Recursos

Em um modelo de controle de acesso baseado em políticas, uma camada de controle de acesso se integra à arquitetura de gestão de identidade inicial exposta na Figura 1. Idealmente, esse modelo deve ser genérico, de forma que seja possível incorporá-lo a ambientes e cenários de organizações virtuais, como *testbeds* de experimentação para Internet do Futuro e cenários de *grid*, além de computação em nuvem.

[4]Vale ressaltar a presença de tais entidades na padronização do ABAC em 2014, modo *enterprise* [Hu et al. 2014]

Na proposta deste trabalho, o controle de acesso baseado em políticas deverá ser integrado a um mecanismo de autorização baseado em um provedor de atributos. O provedor de atributos deverá armazenar os atributos complementares à federação de identidade (por exemplo, CAFe) para um dado usuário. Atributos complementares são aqueles empregados apenas em um contexto específico, como o de um projeto de experimentação em redes. Nesse ambiente é necessário pensar em soluções para, por exemplo, realizar o vínculo entre o identificador do usuário na federação e no provedor de atributos, sendo que esse identificador deve ser o mais opaco quanto possível, a fim de preservar a privacidade do usuário. Este ambiente, onde grupos são formados e têm suas identidades gerenciadas de forma a usufruir de recursos particulares, é conhecido como organização virtual (OV) [Foster et al. 2001].

Na Figura 2, é ilustrada a proposta de integração do mecanismo de autenticação federada, com o mecanismo de autorização baseado em atributos e utilização de políticas de acesso. A figura mostra as entidades da arquitetura proposta envolvidas nas transações de controle de acesso baseado em políticas com a utilização de um provedor de atributos[5].

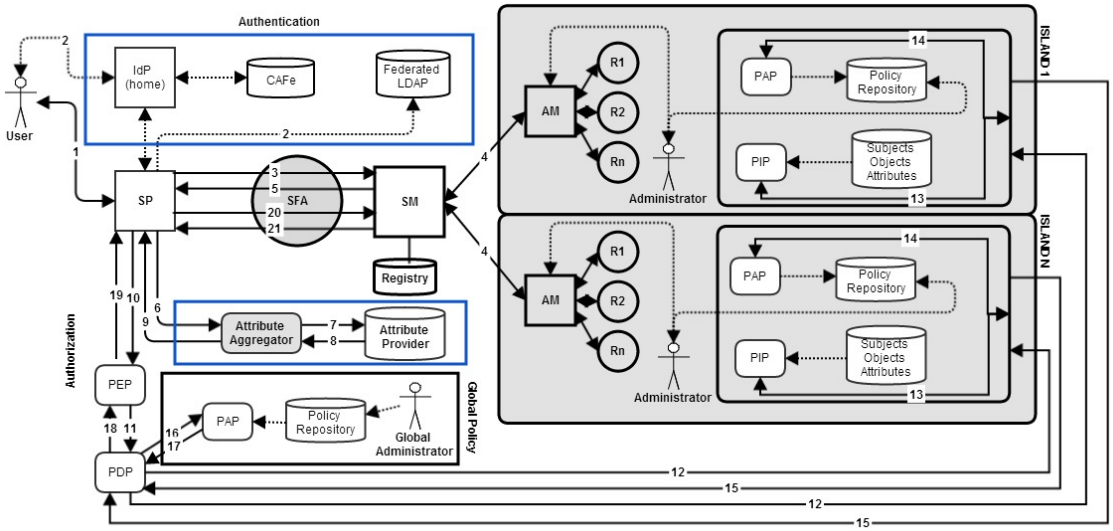


Figura 2. Arquitetura da proposta de controle de acesso baseado em políticas com agregação de atributos.

Utilizando como estudo de caso o projeto FIBRE, descrevem-se os passos realizados desde a autenticação federada até a autorização da utilização de recursos controlados pelas políticas de acesso distribuídas, considerando políticas tanto locais quanto globais. Sendo assim, no passo 1, o usuário acessa o provedor de serviço (SP) que o encaminhará (passo 2) à autenticação, seja ela através da CAFe ou do LDAP FIBRE federado[6]. Tais etapas são tradicionalmente usadas na criação de uma sessão SAML [OASIS 2005], a partir do acesso do usuário ao SP, redirecionamento por meio do WAYF (*Where Are You From*) a seu IdP de origem, autenticação e troca de atributos. Já no passo 3, o usuário, após autenticado, requisita a lista de recursos disponíveis à federação de recursos SFA [Peterson et al. 2010]. Neste passo, o SFA é responsável por comunicar-se com o SM

[5]Devemos destacar que o provedor de atributos adicionais neste trabalho é um diretório LDAP que armazena apenas os atributos adicionais do usuário.

[6]O LDAP FIBRE Federado é um LDAP cuja árvore interliga tanto as instituições brasileiras quanto europeias e suas respectivas ilhas, possibilitando um acesso federado àquele usuário que participa do projeto.

(*Slice Manager*) que tem uma visão global de todos os AM (*Aggregate Manager*) que, por sua vez, tem contato direto com os recursos das ilhas em questão. Sendo assim, os recursos são listados por meio de arquivos do tipo XML, chamado de RSpecs (*Resource Specification*) [Peterson et al. 2010] e retornam até o usuário no passo 5. A partir desse momento, o usuário poderá requisitar os recursos que deseja. Para verificar se o usuário pode ter acesso aos recursos que está solicitando, os passos adicionais estão relacionados ao controle de acesso baseado em políticas e a agregação de atributos que este trabalho propõe. No passo 6, é enviado, pelo SP ao agregador de atributos (*Attribute Aggregator*), um atributo opaco, que identifica o usuário no provedor de atributos (*Attribute Provider*), de tal forma que seja possível recuperar os atributos adicionais da OV sem identificar o usuário diretamente no IdP da federação CAFe. Então, os passos 7 e 8 são responsáveis por recuperar tais atributos e permitir que o agregador de atributos os una aos atributos provenientes da CAFe, complementando o conjunto de atributos do usuário necessários ao acesso solicitado.

Com todos os dados necessários, o SP recebe como retorno do agregador de atributos todos os atributos do usuário no ambiente da OV (atributos da CAFe mais atributos adicionais do Provedor de Atributos), no passo 9, e encaminha tanto esses atributos quanto o RSpec (recebido no passo 5) ao PEP no passo 10. O PEP então realiza a conversão dos atributos em uma pontuação de classificação que indicará em qual classe este usuário se encaixa, considerando os valores de seus atributos. Para este passo temos a definição da Seção 3.1, para maiores detalhes. No passo 11, o PEP realiza a conversão dos arquivos RSpec e de pontuação gerado para XACML. No passo 12, o XACML gerado é enviado à ilha do FIBRE (no caso), e, primeiramente, ao PIP, no passo 13, que realizará a associação do XACML com o seu repositório que deve manter atributos adicionais (opcional), os recursos (*objects*) e dados relativos ao usuário (*subject*)[7]. Sendo assim, no passo 14, as políticas XACML às quais o administrador da ilha previamente cadastrou através do PAP são retornadas ao PDP (passo 15). Neste momento uma política global também é verificada, através do passo 16 e a utilização dos algoritmos de combinação de políticas apresentados pelo próprio XACML [Moses 2005], retornando ao PDP, no passo 17, a decisão tomada. Convertendo ao formato original de RSpec no passo 18 e 19, o PEP entrega ao SP quais recursos o usuário poderá alocar.

Na Figura 2, é possível visualizar também, junto ao SP, o LDAP Federado, que é uma base independente da federação CAFe que permite usuários de instituições ainda não participantes da federação CAFe a ter acesso à federação de recursos. Esta base existe atualmente no FIBRE e aparece nesta proposta como forma também de mostrar a generalização quanto a origem dos atributos do usuário.

3.1. Pontuação de Atributos e Recursos

Para classificar os usuários e facilitar a implementação de políticas de acesso distribuídas em diversas instituições, este trabalho propõe um mecanismo de pontuação de atributos, cuja soma determina a classe de um usuário. Cada atributo deverá ter uma pontuação atribuída de forma a generalizar a definição de políticas no contexto do controle de acesso baseado em políticas e atributos. Sendo assim, é apresentado um exemplo para utilização no caso da federação de experimentação do FIBRE.

[7]Sendo que, no caso deste trabalho, tudo estará vinculado a classes e não a usuários específicos, como forma de deixar mais simples a administração do ambiente de controle de acesso distribuído baseado em políticas

Tabela de Pontuação para Atributos				
Atributo	**Opção**	**Valor**	**Peso**	**Total Parcial**
brEduAffiliationType	student	10	3	30
omfAdmin	TRUE	10	2	20
institution	uff	8	1	8
			Total	**58**

Tabela 1. Tabela de Pontuação para Atributos

Tabela de Pontuação para Atributos	
Pontuação	**Nível**
$0 < X \leq 50$	01
$50 < X \leq 100$	02
$100 < X \leq 200$	03

Tabela 2. Tabela de Pontuação para Atributos

Imaginemos que, conforme a Tabela 1, o usuário experimentador tem o total de 58 pontos. Esses pontos serão utilizados no momento de alocação de recursos, sendo que um usuário se encontra no nível de acesso conforme a Tabela 2. Um usuário nível 02, seguindo o exemplo utilizado para a Tabela 1, tem então direito de alocar até 15 máquinas virtuais, conforme a Tabela 3, por exemplo no ambiente do *testbed* OCF do FIBRE.

Tabela de Pontuação para Recursos para Máquinas Virtuais	
VMs	**Nível**
$0 \leq X \leq 5$	01
$0 \leq X \leq 15$	02
$0 \leq X \leq 20$	03

Tabela 3. Tabela de Pontuação para Recursos para Máquinas Virtuais

4. Resultados

4.1. Agregação de Atributos para Organizações Virtuais

Em [Silva et al. 2013a], foi desenvolvida a parte referente à autenticação destacada na Figura 2, com a integração da federação CAFe à federação de experimentação. Já neste trabalho, são apresentados os resultados de um provedor de atributos com a utilização de um agregador de atributos, permitindo que atributos específicos de uma organização virtual possam ser armazenados separadamente dos atributos advindos da federação acadêmica, considerando como caso de estudo a CAFe. Assim, não é necessário alterar o modelo de armazenamento de atributos da federação.

Modelos de agregação de atributos foram estudados com base no trabalho [Chadwick et al. 2010] e optou-se, pelo cenário aqui aplicado, onde a agregação de atributos é implementada com auxílio de um provedor de serviços (*Service Provider* - SP). Porém, os atributos adicionais armazenados no provedor de atributos não devem identificar o usuário aos quais estão associados. Sendo assim, foi criado um identificador único e opaco, de forma a vincular o usuário da federação acadêmica dentro do provedor de atributos da organização virtual (OV). A ideia base para a utilização de um identificador único e opaco surgiu dos estudos sobre a federação acadêmica SWITCH[8], que define

[8]https://www.switch.ch/aai/support/tools/vo-concept/

também este identificador baseado em outro atributo, também único e fixo do usuário na federação[9].

Conforme a Figura 2 da arquitetura proposta, este trabalho implementa os passos do 1 ao 9. Devemos esclarecer que o atributo único e opaco utiliza a combinação de dois atributos comuns e um hash criptográfico, como é possível ver pelas equações:

$$\delta \leftarrow Attr_u(uid) \cup Attr_u(uidNumber) \quad (1)$$

$$Attr_U(opaque) \leftarrow hash(\delta) \quad (2)$$

O resultado, por exemplo, para o *uid = esilva@uff* com *uidNumber = 1223* é o valor da operação utilizando-se um *hash md*5 da concatenação dos dois atributos, resultando em $af2ec12ce73cc910358ddb400f4abb74$. É interessante utilizar para este método sempre *hash* criptográficos mais atuais, como SHA-1, SHA-2, SHA-256, etc, por exemplo[10].

Na Figura 3, vê-se o passo a passo de autenticação do usuário na federação CAFeExpresso disponível no GidLab. Onde primeiramente é mostrada a tela do serviço de homologação dos atributos, logo após direcionado o usuário ao WAYF para a escolha de sua instituição (*i.e.* IdP1). Feito isso, o usuário insere suas credenciais (*i.e.* "esilva@uff") e se autentica em sua instituição de origem.

(a) Tela inicial de homologação de atributos. (b) Escolha do IdP para autenticação pelo usuário.

Figura 3. Passos para a autenticação CAFe.

Ainda na Figura 4, vê-se os atributos advindos apenas somente daquele IdP, ou seja, os atributos originais daquele usuário em sua instituição. O que se vê mais à frente é a resposta para a mesma tela de homologação de atributos porém utilizando-se o agregador de atributos e o provedor de atributos para a OV do FIBRE. Na Figura 5, os atributos de homologação (validação do sucesso de autenticação do usuário na CAFe) e também os atributos *Attr_opaque*, *Shib-fibre-userEnable* e *Shib-fibre-omfAdmin* aparecem em destaque. Como anteriormente abordado, o *Attr_opaque* foi aquele responsável pela pesquisa possível dos demais dois atributos do usuário contidos no provedor de atributos, neste exemplo.

5. Considerações Finais e Trabalhos Futuros

Este trabalho apresentou uma arquitetura de controle de acesso baseado em políticas, assim como a utilização de um provedor de atributos, para organizações virtuais acadêmicas

[9]https://wiki.shibboleth.net/confluence/display/SHIB2/ResolverScriptAttributeDefinitionExamples

[10]Na validação do modelo do provedor e agregador de atributos, foi utilizado um *hash* criptográfico MD-5

(a) Autenticação sendo realizada pelo usuário "esilva@uff".

(b) Homologação dos atributos padrão advindos do IdP original.

Figura 4. Passos para a autenticação CAFe e homologação dos atributos recebidos pelo SP.

Figura 5. Homologação com atributos agregados.

baseadas em federações de autenticação e autorização. Como resultados são apresentados, dentro da arquitetura proposta e do estudo de caso do projeto FIBRE, um agregador de atributos e o mecanismo de agregação de atributos por meio de um identificador único e opaco. Apesar de usar o FIBRE como estudo de caso, a proposta é genérica e pode ser aplicada em outros tipos de organização virtual onde haja compartilhamento de recursos distribuídos por usuários advindos de diferentes instituições.

Como próximos passos, tem-se a criação de políticas XACML. Tais políticas serão aplicadas tanto utilizando RBAC quanto ABAC no modelo de validação do projeto. Como

exemplo, tem-se como entrada uma solicitação do usuário/experimentador da listagem de recursos. Esta solicitação se dá a federação de recursos (implementada pelo SFA no FIBRE) por meio de um arquivo baseado em XML chamado RSpec, e assim, serão avaliados os atributos e comparados a políticas XACML pré-estabelecidas, armazenadas no PAP. Nesse caso específico, o usuário que envia suas credenciais e é o AM – *Aggregate Manager*[11] – do SFA. O SFA recebe as credenciais do usuário e gera um RSpec. O RSpec retornado para o usuário será interceptado pelo controle de acesso baseado em políticas proposto nesse trabalho. Assim, o retorno ao usuário será um RSpec resultante da filtragem (ou não) proporcionada pela política definida com o XACML.

6. Agradecimentos

Agradecemos o apoio da RNP através do PGID (Programa de Gestão de Identidade), ao GidLab (Laboratório de Gestão de Identidade) da RNP e à CAPES.

Referências

Chadwick, D., Inman, G., and Klingenstein, N. (2010). A conceptual model for attribute aggregation. *Future Generation Computer Systems*, 26(7):1043 – 1052.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. (2001). Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274.

Foster, I., Kesselman, C., and Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222.

Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., and Cybersecurity, S. (2014). Guide to attribute based access control (abac) definition and considerations.

Jin, X., Krishnan, R., and Sandhu, R. (2012). A unified attribute-based access control model covering dac, mac and rbac. In *Proceedings of the 26th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy*, DBSec'12, pages 41–55, Berlin, Heidelberg. Springer-Verlag.

Moses, T. (2005). eXtensible Access Control Markup Language TC v2.0 (XACML).

OASIS (2005). Security assertion markup language (saml) v2.0.

Peterson, L., Ricci, R., Falk, A., and Chase, J. (2010). Slice-based federation architecture. Technical report.

Silva, E., Muchaluat-Saade, D., and Fernandes, N. C. (2013a). Transposicão de credenciais para uso de testbeds para a internet do futuro. In *SBSeg 2013 - WGID*, Manaus.

Silva, E., Muchaluat-Saade, D., Magalhaes, L., Fernandes, N. C., and Rodriguez, N. (2013b). Gestão de identidade em organizacões virtuais. In *JAI 2013*.

Yavatkar, R., Pendarakis, D., and Guerin, R. (2000). A framework for policy-based admission control.

[11] A entidade que mantém contato direto com os recursos existentes.

Um Estudo Comparativo de Estratégias Nacionais de Gestão de Identidades para Governo Eletrônico

Glaidson Menegazzo Verzeletti[1,2], **Michelle Silva Wangham**[1], **Emerson Ribeiro de Mello**[2], **José Alberto Sousa Torres**[3]

[1]Universidade do Vale do Itajaí (UNIVALI) – SC – Brasil

[2]Instituto Federal de Santa Catarina – SC – Brasil

[3]Ministério da Justiça – DF – Brasil

{glaidson.verzeletti, mello}@ifsc.edu.br,

wangham@univali.br, alberto.torres@mj.gov.br

Resumo. *A adoção de programas de Governo Eletrônico (e-Gov) é uma importante ferramenta para promover a transparência dos gastos públicos e o acesso eficiente aos serviços. Para muitos países, é fundamental conceber sistemas de gestão de identidades que ofereçam a autenticação única e o acesso seguro dos cidadãos as aplicações de e-Gov. O presente trabalho descreve e analisa as estratégias nacionais de gestão de identidade dos dez primeiros países do ranking da ONU sobre e-Gov. Por fim, é apresentado um comparativo sobre estas estratégias, destacando as características e soluções comumente adotadas.*

Abstract. *The adoption of e-Gov programs is an important tool for promoting transparency of public expenditure and efficient access to services. To many countries, it is fundamental to conceive Identity Management systems (IdM) that offer single sing-on and also secure access to e-Gov applications by citizens. This paper describes and analyzes national IdM strategies of the top 10 countries on the The United Nations E-Government Survey. Finally, we describe a comparative analysis of these national strategies, highlighting the characteristics and solutions commonly adopted by these countries.*

1. Introdução

O desenvolvimento de programas de Governo Eletrônico (e-Gov) tem como princípio a utilização das tecnologias de informação e comunicação (TICs) para democratizar o acesso à informação, ampliar discussões, promover a transparência e responsabilização das ações e gastos públicos e dinamizar a prestação de serviços públicos, com foco na eficiência e efetividade das funções governamentais [Dawes and Pardo 2002]. Segundo [United Nations 2014], o governo eletrônico é uma importante ferramenta para revitalizar a administração pública tanto no nível nacional quanto local. Nos programas de e-Gov, as colaborações podem ser de cinco formas: entre organizações públicas (G2G), entre organizações públicas e o terceiro setor, entre organizações públicas e privadas (G2B), entre o governo e o cidadão (G2C) e entre governo e seus funcionários (G2E).

Um ponto chave para os programas nacionais de e-Gov é a criação de um sistema de identificação, de autenticação e de autorização de usuários. Esses sistemas são conhecidos como sistemas de gestão de identidades (*Identity Management Systems* – IdM)

[Baldoni 2012]. A gestão de identidades pode ser entendida como o conjunto de processos e tecnologias usados para garantir a identidade de uma entidade, garantir a qualidade das informações de uma identidade (identificadores, credenciais e atributos) e para prover procedimentos de autenticação, autorização, contabilização e auditoria [ITU 2009].

Possibilitar que as aplicações de e-Gov tenham suporte ao processo de autenticação única (*Single Sign-On* – SSO) de usuários é uma facilidade de interesse de muitos países. Porém, boa parte das instituições do governo ainda não formalizaram este processo e, por isto, duplicam o cadastro de pessoas já registradas. A privacidade dos cidadãos é um outro problema a ser tratado nas estratégias nacionais de IdM [Hansen et al. 2008]. Em um cenário ideal, os usuários devem exercer o direito de determinar como suas informações serão manipuladas, quais atributos poderão ser compartilhadas com terceiros, como esse compartilhamento deve ser feito e o período de tempo que essas informações ficarão disponíveis nos sistemas.

Os sistemas de gestão de identidades são complexos, com características poderosas, porém, com algumas vulnerabilidades que podem ser exploradas. Garantir a segurança sem comprometer a privacidade dos usuários e os requisitos não funcionais de usabilidade e de desempenho é um grande desafio no projeto destes sistemas e na concepção de uma estrategia nacional de IdM [Dhamija and Dusseault 2008].

Segundo a Organização para Cooperação e Desenvolvimento Econômico (*Organisation for Economic Cooperation and Development* – OECD), vários países já iniciaram alguma ação em relação à gestão de identidades. As ações de IdM de dezoito nações que fazem parte da OECD foram descritas e analisadas em um relatório [OECD 2011] que apresenta a visão, políticas e estratégias nacionais para gestão de identidades.

Este artigo temo como objetivo descrever e analisar as estratégias nacionais de gestão de identidades dos dez primeiros países do ranking da ONU de Governo Eletrônico [United Nations 2014]. A Seção 2 apresenta as principais características do países analisados e alguns conceitos sobre gestão de identidades e sobre ações de IdM. Uma descrição das estratégias de IdM dos dez países são apresentadas na Seção 3 e uma análise comparativa destas estratégias é descrita na Seção 4. Por fim, na Seção 5, são apresentadas as considerações finais.

2. Aspectos sobre Gestão de Identidade em Programas de e-Gov

A cada dois anos o Departamento de Assuntos Econômicos e Sociais da ONU conduz uma pesquisa sobre o desenvolvimento do e-GOV dos 193 Estados membros. O relatório gerado serve como ferramenta para identificar os pontos fortes e desafios dos programas nacionais e para orientar as políticas e estratégias de e-Gov. A publicação também destaca as novas tendências, questões e práticas inovadoras, bem como os desafios e oportunidades de desenvolvimento de e-Gov [United Nations 2014].

A Tabela 1 apresenta os dez primeiros países do rank da ONU de 2014, suas posições na pesquisa anterior, seus respectivos continentes, os seus índices de desenvolvimento de e-Gov (EGDI), as suas posições no rank que avalia o grau de participação dos cidadãos nas aplicações de e-Gov e no rank de serviços *on-line* que estes países oferecem.

Identidade pode ser definida como um conjunto de dados que representam uma entidade dentro de um determinado contexto. Alguns destes dados podem identificar

Tabela 1. Características dos Países Analisados

Rank (2012)	Rank (2014)	País	Região	EGDI (2014)	Rank e-Particip.	Serviços online
1	1	Coréia do Sul	Asia	0,9462	2	3
12	2	Austrália	Oceania	0,9103	7	7
10	3	Singapura	Asia	0,9076	10	2
6	4	França	Europa	0,8938	4	1
2	5	Holanda	Europa	0,8897	1	8
18	6	Japão	Asia	0,8874	5	4
5	7	EUA	Américas	0,8748	9	5
3	8	Reino Unido	Europa	0,8695	6	10
13	9	Nova Zelândia	Oceania	0,8644	20	14
9	10	Finlândia	Europa	0,8449	25	18

unicamente uma entidade (p.ex. número do CPF) e outros não (p.ex. data de nascimento) [Wangham et al. 2010].

Um sistema de gestão de identidades consiste na integração de tecnologias, políticas e processos de negócio, resultando em um sistema de autenticação de usuários aliado a um sistema de gestão de atributos. [Bhargav-Spantzel et al. 2007] classificam os sistemas de gestão de identidades (IdM) em quatro modelos: tradicional ou isolado, centralizado, federado e centrado no usuário.

No modelo tradicional, tarefas de autenticar usuários (*Identity Provider* – IdP) e prover serviço (*Service Provider* – SP) são realizadas por um mesmo servidor. No modelo centralizado, as tarefas de IdP são realizadas por um único servidor dentro de um domínio administrativo. O provimento de serviços é realizado por um ou mais SPs, os quais possuem relações de confiança com o IdP, garantindo assim que as identidades de usuários são válidas somente dentro deste domínio administrativo. Nos modelos federado e centrado no usuário, também existe uma separação dos papéis de autenticar e prover um serviço. Contudo, as relações de confiança entre IdPs e SPs ultrapassam os limites de domínios administrativos. Isto permite que usuários de uma determinada instituição possam acessar serviços oferecidos em domínios administrativos diferentes [Wangham et al. 2010].

A transposição de informações sobre identidades de usuários de um domínio para outro só é possível se houver uma linguagem padrão para expressar estes dados em ambos os domínios. Em 2005, a OASIS lançou um conjunto de especificações para a troca dinâmica de asserções de segurança baseada no XML. A *Security Assertion Markup Language* (SAML) [OASIS 2005] foi concebida para permitir a troca de informações de autenticação e autorização e garantir o conceito de autenticação única (SSO).

Para muitos países, o desenvolvimento de uma estratégia nacional de IdM é fundamental para a realização do e-Gov. Como estratégia, muitos indicam a necessidade de oferecer serviços com processos de autenticação que exijam credenciais de segurança robustas. A adoção de um sistema de IdM comum permite harmonizar a gestão de identidades em nível nacional. Isto implica em reduzir ou limitar o número de identidades que cada cidadão precisa ter para interagir com os diversos serviços oferecidos pelo governo.

Em suma, grande parte das estratégias dos países buscam reduzir o número de

contas que seus cidadãos precisarão gerenciar e até minimizar a quantidade de vezes que precisarão passar pelo processo de autenticação para ter acesso aos serviços. As soluções adotadas, ou que estão em estudo, geralmente partem da ideia de evoluir práticas e regulamentos usados na identificação tradicional, também chamada de *off-line*.

Segundo [OECD 2011], as políticas descrevem um conjunto de ferramentas que possibilita a implantação da estratégia. As políticas sobre registro dos cidadãos indicam como estabelecer e ligar as identidades eletrônicas com cada cidadão. O processo de registro pode ser centralizado, em países onde a administração pública local é menos autônoma; descentralizado ou federado, em países que dão mais autonomia para a administração local de cada região.

O relatório da [OECD 2011] indica que as políticas que possibilitam a adoção de identidades digitais podem ser voluntárias ou obrigatórias e a escolha entre uma destas está diretamente relacionada com a forma com que cada país opera seus meios de identificação *off-line*.

3. Estratégias Nacionais de Gestão de Identidades

Esta seção apresenta a situação sobre o desenvolvimento e implantação de estratégias nacionais para a gestão de identidade.

3.1. Coreia do Sul

A Coreia do Sul segue o modelo de gestão de identidade centralizada, sendo que todos os cidadãos coreanos possuem um Número de Registro de Residente (RRN) único. O RRN é composto por 13 dígitos e inclui informações como a data e local de nascimento. Desde sua implantação, o RRN tem sido amplamente utilizado em sistemas *on-line*, tanto para interações com o setor público quanto com o setor privado [OECD 2011].

Desde 1999, a estratégia de IdM coreana incentiva o uso de credenciais digitais baseadas em Infraestrutura de Chave Pública (ICP) e, desde 2005, promove o uso de um identificador digital seguro (*i-Personal Identification Number* – i-PIN), tendo como base o RRN. Este sistema de identificação pessoal foi desenvolvido para resolver problemas de segurança relacionados ao roubo de identidade e ao crescente aumento das violações de privacidade e crimes *on-line* [OECD 2011].

3.2. Austrália

O gerenciamento de identidade na Austrália é baseado em uma política de cadastramento descentralizada, sendo um dos principais pontos da estratégia nacional de segurança de identidade manter e tentar fortalecer as credenciais atualmente utilizadas. Documentos de prova de identidade, como passaporte ou carteira de motorista, são emitidos por departamentos específicos sem que haja a necessidade legal de interoperabilidade entre estes sistemas.

Na falta de um identificador único nacional, o documento de boas práticas em e-autenticação australiano permite às agências a utilização dos modelos em silo, centralizado e federado no provimento de autenticação para os seus serviços online.

Em 2007, foi criado o Serviço Nacional de Verificação de Documentos (*Document Verification Service* – DVS), com o intuito de ser usado por órgãos do governo e, potenci-

almente, pelo setor privado. Este serviço permite que agências do governo possam verificar se um documento apresentado pela pessoa foi realmente emitido pelo órgão de origem e se o mesmo foi cancelado ou roubado. Passaportes, vistos e carteiras de motorista são alguns exemplos de documentos que podem ser verificados pelo DVS [OECD 2011]. Em maio de 2014, o procurador geral anunciou o lançamento do DVS comercial, na conferência CeBIT[1] na Austrália. De forma rápida, segura e confiável este produto comercial está sendo expandido para o setor privado, o que permitirá às empresas proteger-se contra os crimes de identidade [Australian Government 2014].

3.3. Singapura

Singapura iniciou seu projeto de e-Gov na década de 80 com o objetivo de transformar o governo em um modelo mundial em termos de tecnologia da informação. No final dos anos 90, houve uma convergência das políticas de TIC o que permitiu abrir o caminho para a criação do plano de ação e-Gov I (2000-2003) e para o plano de ação II (2003-2006). O principal objetivo do primeiro plano era criar o maior número possível de serviços públicos *on-line*, enquanto a ênfase para o segundo foi melhorar a experiência dos usuários no uso dos serviços [Infocomm Development Authority of Singapore 2014].

Fatores como a alta renda per capita (US$ 47,210)[2], população pequena e a entrada dos dispositivos móveis no mercado, favoreceram o desenvolvimento e o acesso aos sistemas de governo, principalmente nas modalidades G2C e G2B. Desde de 2003, todos os residentes de Singapura com idade igual ou superior a quinze anos podem fazer uso de uma credencial única para realizar transações nos diferentes sistemas do governo, serviço este denominado *SingPass ID/password* [Infocomm Development Authority of Singapore 2014].

Desde 2009, empresas, sociedades, instituições de saúde, sindicatos, entre outros, que estão registradas em Singapura, passaram a utilizar uma identificação única (*Unique Entity Number - UEN*) para as interações com o governo. Dentre os benefícios trazidos pela UEN[3] para as entidades, estão a facilidade na apresentação de declarações fiscais, envio de contribuições de empregados e a aplicação de licenças de importação e exportação [Singapore Government 2008]. Atualmente, cidadãos e empresas podem acessar mais de 1.600 serviços *on-line* e mais de 300 serviços providos pelo governo da Singapura [IDA Singapore 2014].

3.4. França

Na França, todo provimento de serviços *on-line* para os cidadãos e empresas é feito a partir de um portal do governo[4], sendo que o processo de autenticação exige certificados digitais emitidos por provedores de serviços de certificação (CSPs) qualificados pelo governo e avaliados em função dos requisitos exigidos pelo "Framework Geral de Segurança" (*Référentiel Général de Sécurité - RGS*) [European Comission 2014b]. O RGS provê atualmente três níveis garantia de segurança (*Level of Assurance - LoA*): elementar[5], padrão

[1] http://www.cebit.com.au/conferences
[2] Singapore Police Force, 2013. Disponível em: http://www.spf.gov.sg/sms70999
[3] http://www.uen.gov.sg
[4] www.service-public.fr
[5] Nível Elementar: assinatura pode ser armazenada em um módulo de software.

e reforçado[6] [France Government 2013].

Em 2005, o governo da França iniciou o projeto (*Identité Nationale Electronique Sécurisée* – INES) com o intuito de criar um cartão de identidade eletrônico. O cartão eID contém informações pessoais, como por exemplo nome completo, data de nascimento e endereço, além de guardar informações que podem ser usadas em processos de autenticação mais robustos, como informações biométrica, certificado digital e assinatura eletrônica [European Comission 2014b].

O governo optou por não tornar obrigatório a adoção deste cartão para seus cidadãos, porém, de acordo com o plano de desenvolvimento para a economia digital de 2012, o governo Francês pretende fazer uso deste cartão para permitir aos seus cidadãos participar de processos de decisão pública [European Comission 2014b]. Desta forma, apesar de não ser obrigatório, o governo espera que a população busque pelo cartão, uma vez que o mesmo lhe dará direito a voz nos processos de decisão do governo.

3.5. Holanda

A estratégia holandesa de IdM está baseada no DigiD[7], um mecanismo nacional de identidade e autenticação digital para transações eletrônicas entre cidadãos e empresas com órgãos públicos [OECD 2011].Atualmente, o sistema oferece duas formas para realizar a autenticação: *DigiD Basic* – que faz uso somente de nome de usuário e senha; *DigiD Medium* – que além do nome de usuário e senha também exige uma verificação por meio de mensagens SMS.

O objetivo é que o DigiD se torne o sistema de autenticação utilizado na administração pública para prestar serviços eletrônicos aos cidadãos. Apesar da adoção não ser obrigatória, mais de 9,8 milhão de holandeses já ativaram a sua conta DigiD, que pode ser utilizada em mais de 600 organizações governamentais ou empresas privadas que executam serviços públicos [OECD 2011].

O governo está trabalhando em um programa chamado "eRecognition para empresa" com o objetivo de permitir que o DigiD seja usado também nas interações com empresas privadas (G2B). O eRecognition oferecerá diferentes mecanismos de autenticação, desde combinações de nome de usuário e senha até soluções baseadas em ICP [European Comission 2014c].

3.6. Japão

Atualmente todo cidadão japonês deve se cadastrar no sistema de Registro de Residente Básico, fornecendo aos governos municipais informações como: nome, data de nascimento, sexo e endereço físico. Em 2002, estas quatro informações começaram a alimentar o sistema JUKI-NET, criado para compartilhar dados entre os órgãos governamentais, nascendo assim o modelo centralizado de gestão de identidade no país. Este sistema tem como base os dados registrados em 3.200 municípios, oferecendo aos cidadãos a opção de obter o cartão de identificação (*My Number*)[8], o qual faz parte da estratégia nacional de oferecer uma identificação única a todo cidadão a partir de 2015 [Rebecca Bowe 2012].

[6]Nível Padrão e Reforçado: a chave de assinatura é armazenada em um dispositivo criptográfico de hardware, como um *smart card* ou uma chave USB.

[7]https://www.digid.nl

[8]Cartão com chip, que contém as informações de registro obrigatórias, foto e número de identificação.

Segundo projeto de lei aprovado em 2013 pela Câmara dos Deputados, a partir de 2016 todo cidadão japonês deverá possuir um cartão *My Number*. Este cartão será usado para compartilhar informações entre as agências que administram seguro social, impostos e programas de mitigação de desastres [Yumi Watanabe 2014]. Com o objetivo de expandir o uso para outras áreas, em 2018 o processo passará por uma avaliação.

3.7. Estados Unidos

Em 2003, os Estados Unidos decidiram adotar o modelo de identidades federadas, voltado para órgãos públicos ou entidades da iniciativa privada. A federação é baseada em quatro níveis de garantia e o cidadão, ao tentar acessar um serviço governamental, é direcionado para uma lista de provedores de identidade que possuem a garantia necessária para acesso àquele serviço específico. Em 2009, cerca de 27 agências americanas já proviam os seus serviços com base na federação [Seltsikas and van der Heijden 2010].

O conceito chave adotado pelo governo dos Estados Unidos é a participação voluntária de indivíduos e de organizações. Segundo o [OECD 2011], o governo não impõe o aceite de soluções específicas, p.e. uso de certificados digitais para autenticar pessoas ao realizarem transações *on-line* com o governo. As políticas específicas de segurança para a estratégia de IdM *on-line* ainda estão sendo escritas, entretanto, tem-se como diretrizes de segurança o uso de criptografia forte, padrões abertos, como por exemplo o SAML, e a adoção de sistemas de informações que possam ser auditáveis [OECD 2011].

3.8. Reino Unido

O portal *Website Government Gateway* permite aos cidadãos fazerem seu registro inicial, fornecendo informações pessoais, além de sua senha. Como resultado, o cidadão recebe um código de segurança (PIN) por meio de correspondência em sua residência. Este código é então usado pelo cidadão para usufruir de alguns dos serviços *on-line* oferecidos pelo governo. Alguns serviços, por serem considerados mais críticos, podem possuir mecanismos de autenticação mais rígidos, exigindo por exemplo, autenticação biométrica ou por meio de certificados digitais [European Comission 2014d].

O ano de 2012 marcou uma mudança radical no desenvolvimento do governo eletrônico do Reino Unido, já que foi o ano que iniciou o Programa de Garantia de Identidade (IDAP), capitaneado pelo Gabinete do Primeiro Ministro. Por meio deste programa, o governo pretende oferecer um meio mais seguro para os cidadãos provarem a sua identidade ao interagir com os serviços de governo eletrônico.

O modelo utiliza um "hub" que permite que diferentes provedores de identidade autentiquem os indivíduos para os provedores de serviço sem que seja necessário que o governo armazene de forma centralizada os dados pessoais dos usuários e sem que a privacidade seja afetada por trocas desnecessárias de dados ou por compartilhamento indevido dos dados do usuário sem o seu consentimento.

A primeira etapa do programa foi definir quais seriam os provedores de identidade. Para isso, o governo realizou um processo licitatório que terminou com a contratação de cinco empresas - Digidentity, Experian, Mydex, The Post Office e Verizon. É interessante ressaltar que, neste modelo, os próprios provedores de identidade efetuam os cadastros dos usuários, recebendo do governo por cada usuário cadastrado. A segunda etapa é marcada pela difusão da utilização do serviço dos provedores de identidade entre os órgãos governamentais [Government Digital Service 2014].

3.9. Nova Zelândia

Optou-se por uma estratégia de IdM visando acelerar o desenvolvimento e oferta de serviços *on-line* para seus cidadãos. Foi adotada uma política de registro descentralizada, sendo que cada governo local tem autonomia para indicar como registrar seus cidadãos, bem como para indicar quais mecanismos de autenticação podem ser usados nos serviços.

A chave de sucesso para a implantação da solução de IdM foi a adoção do *e-Government Interoperability Framework* (e-GIF), também conhecido como NZ e-GIF, lançado em 2002 [OECD 2011]. O e-GIF possui uma versão própria do SAML (NZ SAML), sendo que a primeira versão do *framework* teve como foco a autenticação e as versões subsequentes em atributos e autorização [Kãwanatanga 2008].

3.10. Finlândia

O "Sistema de Informação da População" é o responsável pelo cadastro nacional dos cidadãos finlandeses e dos cidadãos estrangeiros com residência permanente no país, o qual é mantido pelo Centro de Registro da População (CRP) e pelos cartórios locais. O CRP é a única autoridade certificadora na Finlândia capaz de realizar a emissão de certificados Pan-Europeus [European Comission 2014a], sendo o responsável pela emissão de identidades eletrônicas (eID) e dos certificados digitais para os cidadãos (FINeID[9]). Somente a partir destas credenciais, é possível acessar os serviços de e-Gov.

4. Comparação e Análise das Estratégias Nacionais

Segundo a [OECD 2011], os países encontram-se em diversos estágios em relação ao desenvolvimento e implementação das estratégias nacionais de IdM. A partir do **desenvolvimento de políticas** (definição de leis, planos, ações, etc), os governos conseguem **implementar suas estratégias de IdM**. Na Tabela 2, são apresentados os países de acordo com o estágio de desenvolvimento e implementação das suas estratégias.

Tabela 2. Status estimado para as Estratégias Nacionais de IdM [OECD 2011]

ESTÁGIO	DESENVOLVIMENTO	IMPLEMENTAÇÃO
Não iniciado	Japão	Japão, Estados Unidos
Estágio Inicial	Estados Unidos	Austrália, Nova Zelândia
Em Andamento		Coreia do Sul, Holanda
Estágio Final	Austrália	
Totalmente Desenvolvida	Coreia do Sul, Nova Zelândia	

Alguns países, como o Japão por exemplo, mostram sinais de evolução em relação ao estágio de desenvolvimento e implementação de suas estratégias, podendo ser classificado como "em andamento" e "estágio inicial", respectivamente. Porém, esta classificação só poderá ser realmente reavaliada entre 2015 e 2016, após as políticas do "*My Number*" serem de fato colocadas em prática. Por outro lado, a Coreia do Sul está com suas estratégias totalmente desenvolvidas e avança para o estágio final de implementação. Ainda de acordo com [OECD 2011], estes países procuram focar suas estratégias de Governo Eletrônico na administração pública, esperando que estas sejam adotadas pelo setor privado. Vale destacar que Singapura, França e Finlândia, foram países que não participaram da pesquisa da OECD e, por isto, o estágio de suas estratégias não estão indicados acima.

[9]Sistema de certificados do CRP, baseado em uma Infraestrutura de Chave Pública.

Tabela 3. Comparativo entre países

PAÍS	MODELO IdM	Id ÚNICO	SAML 2.0	Participação E. Privadas	Participação Cidadão
Coréia do Sul	Centralizado	Sim	Sim	Sim	Obrigatória
Austrália	Federado	Não	Sim	Sim	Voluntária
Singapura	Centralizado	Sim	-	Sim	Voluntária
França	Centralizado	Sim	-	Sim	Voluntária
Holanda	Centralizado	Sim	Sim	Sim	Voluntária
Japão	Centralizado	Sim	-	Não	Obrigatória
EUA	Federado	Não	Sim	Sim	Voluntária
Reino Unido	Federado	Sim	Sim	Não	Voluntária
Nova Zelândia	Federado e Centrado no usuário	Não	Sim	Não	Voluntária
Finlândia	Centralizado	Sim	-	Sim	Voluntária

A Tabela 3 resume e compara algumas características das estratégicas de gestão de identidade dos países analisados. Pode-se observar que a maioria dos países adota o SAML como padrão e, normalmente, é utilizado um identificador único para o acesso aos sistemas. A participação de entidades privadas e não-governamentais é geralmente incentivada, muito embora se observa nos países que não têm estratégias implementadas para este segmento, algumas iniciativas do governo em estender as políticas de G2B.

Embora não seja uma regra, nota-se a adoção de modelos de IdM centralizado ou federado, sendo que a participação do cidadão é normalmente voluntária. Além disso, o governo não impõe soluções específicas, como por exemplo, o uso de certificados digitais.

5. Conclusões

O governo brasileiro encontra-se na posição 54 do ranking da ONU [United Nations 2014] e ainda não definiu a estratégia nacional de gestão de identidades para e-Gov. Existe apenas uma definição de padrões de interoperabilidade de sistemas (arquitetura e-PING) [BRASIL 2014]. Dentre estas, destacam-se o uso do SAML como padrão para a troca de informação sobre autenticação e autorização entre domínios, da especificação WS-Security 1.1 para o fornecimento de segurança às mensagens trocadas e WS-Trust 1.4 para a gestão das relações de confianças (intermediação).

Para conceber uma estratégia nacional de IdM para o governo brasileiro, é muito importante analisar as estratégias adotadas nos países que se destacam na provimento de e-Gov, porém sem esquecer das peculiaridades do país, tais como a sua dimensão territorial, o índice de inclusão digital e o elevado índice de fraldes eletrônicas. Como trabalhos futuros, pretende-se aprofundar a análise considerando outros aspectos de gestão de identidades e aumentar o número de países analisados para os vinte melhores do rank da ONU.

Referências

Australian Government (2014). Identity security. http://goo.gl/9oy8EC.

Baldoni, R. (2012). Federated identity management systems in e–government: the case of italy. *Electronic Government, an International Journal*, 9(1):64–84.

Bhargav-Spantzel, A., Camenisch, J., Gross, T., and Sommer, D. (2007). User centricity: a taxonomy and open issues. *Journal of Computer Security*, 15(5):493–527.

BRASIL (2014). e-ping padrões de interoperabilidade de governo eletrônico. Technical report, Comitê Executivo de Governo Eletrônico. http://goo.gl/PsV0UT.

Dawes, S. and Pardo, T. (2002). Building collaborative digital government systems. In *Advances in Digital Government*, volume 26, pages 259–273. Springer US.

Dhamija, R. and Dusseault, L. (2008). The seven flaws of identity management: Usability and security challenges. *Security Privacy, IEEE*, 6(2):24–29.

European Comission (2014a). eGovernment in Finland. eGovernment Factsheets.

European Comission (2014b). eGovernment in France. eGovernment Factsheets.

European Comission (2014c). eGovernment in the Netherlands. eGovernment Factsheets.

European Comission (2014d). eGovernment in the U.K. eGovernment Factsheets.

France Government (2013). TSL and RGS. http://goo.gl/vgqjat.

Government Digital Service (2014). Identity Assurance: First delivery contracts signed. http://goo.gl/7El4Ys.

Hansen, M., Schwartz, A., and Cooper, A. (2008). Privacy and identity management. *Security Privacy, IEEE*, 6(2):38 –45.

IDA Singapore (2014). egov2015 masterplan (2011-2015) - visionstrategic thrusts. http://goo.gl/Yzqx8v.

Infocomm Development Authority of Singapore (2014). egov masterplans. http://goo.gl/Hrw8D2.

ITU, T. (2009). Series y: Global information infrastructure, internet protocol aspects and next-generation networks. *Rec. ITU-T Y*, 2720.

Kãwanatanga, T. K. O. N. T. (2008). *New Zealand E-government Interoperability Framework (NZ e-GIF)*. State Services Commission.

OASIS (2005). *Security Assertion Markup Language (SAML) 2.0 Technical Overview*.

OECD (2011). National strategies and policies for digital identity management in OECD countries. *OECD Digital Economy Papers*, (177).

Rebecca Bowe (2012). In japan, national ID proposal spurs privacy concerns. http://goo.gl/jTfLk6.

Seltsikas, P. and van der Heijden, H. (2010). A taxonomy of government approaches towards online identity management. In *43rd HICSS*, pages 1–8.

Singapore Government (2008). Unique entity number brings convenience to entities. http://goo.gl/mPCVSG.

United Nations (2014). e-Government Survey: E-Government for the Future We Want. Economy & Social Affairs.

Wangham, M. S., de Mello, E. R., da Silva Böger, D., Gueiros, M., and da Silva Fraga, J. (2010). *Minicursos X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, chapter Gerenciamento de Identidades Federadas, pages 1–52.

Yumi Watanabe (2014). ANALYSIS: Japanese law to establish new ID number system includes measures to address privacy concerns. http://goo.gl/TspeMG.

Diagnóstico do governo eletrônico brasileiro – uma análise com base no modelo de gerenciamento de identidades e no novo guia de serviços

José Alberto Sousa Torres [1], Flávio Elias Gomes de Deus[2], Rafael Timóteo de Sousa Júnior [2]

[1]Departamento de Ciência da Computação– Universidade de Brasília (UnB)
Caixa Postal 4.466 – 70.910-900 – Brasília – DF – Brasil

[2]Departamento de Engenharia Elétrica – Universidade de Brasília (UnB)
Caixa Postal 4.386 – 70.910-900 – Brasília – DF – Brasil

alberto@torres.eti.br, flavioelias@unb.br, desousa@unb.br

Abstract. *This article aims to provide a diagnosis of the Brazilian electronic government, emphasizing the relationship of e-government with the national strategy for identity management and the analysis of the new public services guide - the portal www.servicos.gov.br.*

Resumo. *Este artigo tem como principal objetivo fazer um diagnóstico comparado do governo eletrônico brasileiro, dando ênfase na relação do e-gov com a estratégia nacional de gerenciamento de identidade e na análise do novo guia de serviços públicos – o portal www.servicos.gov.br.*

1. Introdução

Nos últimos anos, o impacto da tecnologia na sociedade está aumentando consideravelmente, tendo a Internet se tornado uma das principais ligações entre a população e o mundo digital. Com a influência da rede mundial de computadores, as pessoas têm adotado novas formas de se comunicar com o mundo, o que as tem levado a esperar que informações e serviços estejam disponíveis online e acessíveis a qualquer hora e lugar, o que levou, na última década, uma considerável quantidade de governos a adotar ações para publicar os seus serviços e informações na web [Urdiales 2004].

O conceito de governo eletrônico (e-gov) abarca justamente esta iniciativa, do uso da tecnologia da informação como forma de entregar serviços e informações governamentais aos cidadãos, empresas ou outros governos, através da integração dos processos [Alemayehu & Mwangi. 2011]. Nesta linha, tópicos como a migração dos serviços públicos para o ambiente online, o estímulo à utilização de serviços eletrônicos por parte dos cidadãos, a publicação dos dados governamentais em portais públicos e a implantação de estratégias para prover confiança na troca de informações têm sido tratados como diretrizes da agenda de governo eletrônico em muitos países.

No Brasil, o tema da informatização dos serviços do governo vem sendo pauta desde a década de 70. A partir de então, e por quase duas décadas, o modelo brasileiro se traduziu na montagem de empresas estatais de serviços de processamento de dados, como o Serpro e Dataprev [Takahashi 2000]. A mudança de paradigma se iniciou em

1993, quando surgiu um movimento de alguns ministérios no sentido de utilizar a Internet para divulgar informações armazenadas em seus bancos de dados. Deste então, tem se observado um avanço importante do e-gov brasileiro, que culminou com o lançamento do Guia de Serviços Públicos do Governo Federal, no ano de 2012.

O principal objetivo deste trabalho é o de apresentar um estudo sobre o governo eletrônico brasileiro, dando ênfase à realização de um diagnóstico do estado atual com base no modelo de gerenciamento de identidades nacional e no novo portal de serviços do governo federal, o *www.servicos.gov.br*.

2. Governo Eletrônico

Nos últimos vinte anos, tem-se observado um desenvolvimento tecnológico acima de todas as expectativas. Apesar dos computadores pessoais já fazerem parte do cotidiano das pessoas desde praticamente o início deste movimento, a onda de popularização da Internet é um fenômeno consideravelmente recente.

Dentro deste curto período de tempo, as facilidades da internet foram associadas ao conceito de globalização, transformando o mundo em uma pequena aldeia global. Este rápido desenvolvimento da rede mundial de computadores e outras mídias digitais encorajaram profissionais, consultores e autoridades do governo a empregar cada vez mais tecnologia no fornecimento de serviços públicos aos cidadãos, criando o que ficaria conhecido como governo eletrônico [Shareef 2012].

Entre as diversas promessas desta revolução digital, uma das mais relevantes é o potencial que o governo eletrônico tem para fortalecer a democracia e tornar a administração pública mais próxima dos cidadãos [InfoDev 2002]. A presença "online" dos governos possibilita a participação popular independente da distância física ou da condição financeira dos indivíduos. Além de se apresentar como uma medida de inclusão social, a chegada do e-gov tem feito com que a máquina pública se torne mais eficiente, auxiliando os governos na redução dos gastos públicos.

Segundo DeBenedictis (2002), "e-gov" é o conjunto de atividades conduzidas digitalmente e que são realizadas pelo governo ou relacionadas ao mesmo. Mais especificamente, governo eletrônico descreve o uso de tecnologias da informação baseadas na Internet para melhorar a performance e a governança do governo, através da melhor, mais rápida e mais barata execução de atividades governamentais, acesso a informações e participação dos cidadãos e organizações nas decisões públicas.

De acordo com Siau & Long (2006), o governo eletrônico pode ser classificado em quatro grandes áreas:

(1) Governo-para-cidadão (G2C) - voltado para o fluxo de comunicação entre o governo e os cidadãos;

(2) Governo-para-empresas (G2B) - visa prover um melhor serviço para as empresas, eliminando a redundância na coleta de dados e reduzindo custos para o governo;

(3) Governo-para-funcionários (G2E) – é relacionado aos serviços prestados pelo governo a funcionários públicos para aumentar a eficiência e eficácia da administração governamental; e

(4)Governo-para-governo (G2G) – que permite um avanço na cooperação e colaboração entre governos de diferentes níveis.

A principal ferramenta para diagnóstico do avanço do governo eletrônico nos países vem sendo os relatórios bianuais de e-gov publicados pela Organização das Nações Unidas. A metodologia utilizada pela ONU para cálculo do índice de desenvolvimento do governo eletrônico (EGDI) se mantem a mesma desde 2001, quando se iniciou o estudo. O EGDI é um índice composto por três dimensões: a provisão de serviços online, a infraestrutura de telecomunicações e o capital humano.

O último diagnóstico, realizado em 2014, demonstrou que quase todos os países avaliados apresentaram melhora no seu índice de desenvolvimento de e-gov, se comparado com o estudo de 2012. Neste ano, pela primeira vez, constatou-se que todos os membros associados à ONU, apesar da maioria ainda permanecer nos níveis mais baixos de desenvolvimento do governo eletrônico, já possuíam um website nacional oficial [ONU 2014].

Outra constatação importante foi o fato de que as nações pobres, localizadas em sua maior parte no continente africano, agregam os índices de EGDI mais baixos, dificilmente ultrapassando o valor de 0,25. Em contraste, os países com alto índice de EGDI (>0,75) estão quase sempre entre as nações desenvolvidas e ricas, em sua maioria situadas nos continentes europeu e norte americano. A localização geográfica dos países com maior índice EGDI pode ser visualizada, em azul, na Figura 1.

Figura 1. Localização dos 25 países com maior índice EGDI [ONU 2014]

2.1. Governo Eletrônico e Gestão de Identidade

Por muito tempo, o gerenciamento de identidades foi o principal desafio técnico para a difusão do governo eletrônico. Desta forma, o avanço recente no desenvolvimento de estratégias nacionais para gerenciamento das identidades digitais trouxe um novo fôlego para o avanço do e-gov.

Em todo o mundo, os países estão criando ou reforçando as suas estratégias de gestão digital de identidade (IdM), tanto pelo crescente interesse na utilização do ambiente on-line para prestação de serviços de governo eletrônico, quanto pela

necessidade de se combater os crimes de roubo de identidade [McKenzie 2008]. Neste contexto, governos vêm atuando como provedores de serviço e criadores de políticas para regular o uso e a proteção destas identidades. Como provedores de serviços, muitos países têm trabalhado em projetos de identidade digital para seus cidadãos e funcionários [Bertino & Takahashi 2011].

A utilização do gerenciamento eletrônico de identidade é um importante caminho pelo qual os governos podem padronizar e unificar o acesso aos seus serviços de e-gov, permitindo o seu fornecimento de forma adaptada às necessidades do usuário. Segundo dados da ONU (2014), o número de países que estão atualmente desenvolvendo ações voltadas à utilização de uma identidade eletrônica centralizada para acesso aos serviços governamentais passou de 52, em 2012, para 69, em 2014. Este estudo apontou, ainda, a importância da implantação destas estratégias de gerenciamento para que os serviços de governo eletrônico consigam alcançar os últimos estágios de desenvolvimento, o transacional e o conectado.

Outra pesquisa, esta realizada pela Organização para Cooperação e Desenvolvimento Econômico – OECD, também confirma este cenário. Realizado com 18 países pertencentes a quatro continentes, o estudo identificou que todos os governos pesquisados já possuem ou estão desenvolvendo uma estratégia nacional para gerenciamento digital de identidade. Para a maioria destes países, o primeiro objetivo a ser atingido é o de tornar realidade o governo eletrônico e promover inovação nos serviços públicos e privados, além de aperfeiçoar a arquitetura técnica desenvolvida para promover a interoperabilidade de sistemas. Outro fator observado foi que, na maioria dos países analisados, as estratégias de gerenciamento de identidade já existentes são adaptadas para o mundo digital. [OECD 2011].

3. E-gov e Identidade no Brasil

Na última pesquisa realizada pela ONU (2014), o Brasil estava na posição 57 de desenvolvimento de governo eletrônico no planeta e na 8ª posição do continente americano. Isto indica que o nível de evolução do e-gov em nosso país ainda não é suficiente para posicioná-lo, em um curto espaço de tempo, entre os primeiros colocados do ranking, sobretudo pelo fato de que, dentre os elementos do índice, são consideradas dimensões relacionadas ao avanço social do país, além do fato de que a falta de uma estratégia de gestão de identidade impede o avanço dos e-serviços brasileiros para o nível transacional de maturidade de e-gov.

Segundo os dados do relatório da ONU, apesar de 100% dos serviços brasileiros já terem atingido o estágio de "emergente", apenas 26% dos serviços alcançaram o último estágio, o conectado. Nos estágios intermediários, temos 68% dos serviços classificados como melhorados, e 28% como transacionais, sendo que um serviço pode se enquadrar em mais de um estágio simultaneamente. Estes níveis de maturidade são definidos como uma forma de mensurar o grau de evolução dos serviços eletrônicos, servindo como uma referência para comparação entre a situação de diferentes países.

A relação entre os níveis transacional e conectado com a estratégia de gerenciamento de identidade vem da necessidade oriunda destes níveis de maturidade em se ter uma identificação segura e confiável do usuário que está realizando a operação no ambiente online. E é justamente neste ponto que pode-se identificar um dos

principais problemas brasileiros, já que, apesar do Brasil estar no rol de países que tradicionalmente exigem de seus cidadãos um documento de identificação para permitir o reconhecimento de suas relações com a sociedade e interações com entes públicos e privados, e destarte o avanço do governo eletrônico mundial em direção à criação de uma identidade única digital, não foram encontradas evidências de que o nosso país possua uma Estratégia Nacional de Gerenciamento de Identidades.

Com base na experiência de outros países, o que se observa é que as estratégias de implantação de identidades eletrônicas tomam como base um cadastro populacional previamente existente e normalmente unificado. Por muito tempo, associou-se a inércia do Brasil na definição de sua estratégia de gerenciamento de identidade a inexistência de uma base civil nacional, pelo fato de que o cadastramento da população e a emissão das carteiras de identidade sempre foram realizados de forma descentralizada pelos Estados que, nos termos da nossa atual constituição, são autônomos em relação à União.

No início de 2014, a soma de brasileiros cadastrados nos 27 estados perfazia um total de 244.887.089 cidadãos. Este número, cerca de 20% maior do que a população nacional prevista para 2013, é decorrência da manutenção do cadastro de pessoas falecidas, além do fato de que é permitido ao cidadão brasileiro tirar uma identidade em cada estado, ou seja, uma única pessoa pode se cadastrar 27 vezes. Destes registros, apenas 25,3%, ou 61.982.449, se encontravam inseridos em um sistema automatizado de de-duplicação biométrica [BRASIL 2014].

Esta realidade deveria ter mudado com a promulgação da Lei n.º 9.454 em 1997, onde o legislador definiu a criação de um número único "nacional" para cada cidadão brasileiro – o Registro de Identidade Civil (RIC), além de ter dado ao Poder Executivo autorização para definir a entidade gestora central do sistema. Desta forma, os Estados e o Distrito Federal seriam os responsáveis por alimentar a base nacional com os dados locais, ao tempo que passariam a ter acesso a esta base unificada e centralizada. Neste modelo, o órgão central do RIC exerceria o papel de Provedor de Identidade em um modelo de Gerenciamento de Identidades Centralizado. Entretanto, passados dezessete anos do início da vigência da lei 9454, o RIC ainda se encontra em processo de estudo pelo Ministério da Justiça (MJ), muito pelo fato de não constar dentre os projetos da agenda estratégica do governo federal

Esta ausência de um identificador único nacional levou os órgãos federais a adotarem a estratégia em Silo para gestão da identidade. Desta forma, cada entidade construiu a sua própria base de usuários de modo a prover a identificação necessária para fornecimento dos serviços. Um resumo de algumas das maiores bases de dados federais é exibido na Tabela 1.

Tabela 1 – Bases de dados federais

Base de Dados	Órgão Gestor	Tamanho
CPF	Ministério da Fazenda/Receita Federal	Mais de 200 milhões
Passaporte	Ministério da Justiça/DPF	11.974.026
SECAD	Tribunal Superior Eleitoral	142.822.046
Cadastro Único	Ministério do Desenvolvimento Social	86.781.675

3.1. Diagnóstico do portal servicos.gov.br

Com o objetivo de caracterizar mais fielmente o atual estado do e-gov brasileiro e o nível de relação com políticas de gestão de identidade, foi realizada uma análise detalhada no portal servicos.gov.br, que foi criado com o intuito de ser o guia de serviços públicos do governo federal, uma espécie de catálogo eletrônico de serviços.

Para realizar o estudo, os 597 serviços listados no portal, levando em consideração o mês de março de 2014, foram acessados e classificados em relação às seguintes características:

- Provedor do serviço
- Forma de Acesso (Online ou Offline)
- Utilização do certificado digital
- Categoria de governo eletrônico (G2C, G2E, G2B e G2G)

O resultado da análise apontou que os três maiores provedores de serviço do governo eletrônico brasileiro são a Receita Federal, com 440 itens cadastrados, o Ministério da Justiça, com 42, e o Ministério da Previdência Social, com 40. É interessante ressaltar que no portal de serviços, via de regra, os Ministérios agrupam na estrutura hierárquica superior os serviços prestados por órgãos subordinados. A quantidade de serviços associados ao Ministério da Justiça, por exemplo, leva em consideração na contagem todos os serviços prestados pela Polícia Federal e Rodoviária Federal, além dos outros órgãos vinculados ao MJ. Exceção se dá no caso da Receita, onde a listagem dos seus serviços foi realizada de forma independente do Ministério da Fazenda, apesar de estar formalmente vinculada a este. A distribuição dos serviços por seus provedores pode ser visualizada no Gráfico 1.

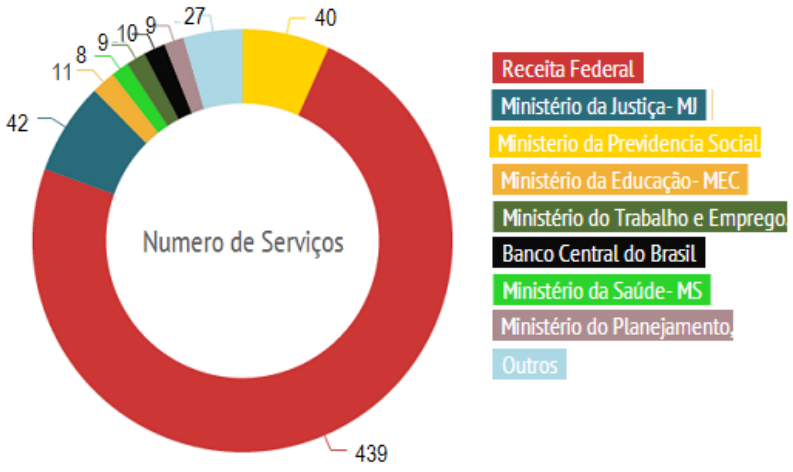


Gráfico 1. Principais provedores do e-gov brasileiro

A diferença observada na quantidade de serviços provida entre o primeiro e o segundo lugar do ranking levou a uma análise mais rebuscada da descrição dos serviços, que culminou em uma proposta de agrupamento de alguns dos serviços prestados pela Receita Federal.

No conceito de Arquitetura Orientada a Serviço (SOA) adotado por Fugita e Hirama (2012), um serviço é a execução de um trabalho ou de uma função de um prestador para um requisitante, agrupando logicamente operações que, fazendo uma analogia com Orientação a Objetos, estão para os serviços como os métodos estão para as classes. Entretanto, quando se observou a lista de serviços da Receita Federal, ficou evidente que diversas operações estavam sendo definidas como serviços. Por exemplo, as operações de Alteração de endereço no CPF, Alteração cadastral no CPF, Comprovante de Situação Cadastral no CPF, Comprovante de Inscrição no CPF, Consulta Informações Cadastrais no CPF e outras vinte operações relacionadas ao CPF acabaram definidas como serviços independentes. O mesmo ocorreu para os serviços relacionados ao CNPJ, Comércio Exterior, entre outros.

Isto explica a diferença de serviços publicados entre a Receita e o Ministério da Justiça, segundo colocado. Fazendo uma reengenharia dos serviços do portal *servicos.gov.br*, levando em consideração a definição de Fugita e Hirama (2012), foi possível agrupar as operações da Receita Federal em 91 serviços, o que ainda a mantém na liderança, mas torna a estatística mais próxima do real. Apesar deste trabalho de reengenharia, continuaremos a trabalhar, no decorrer da análise, com o número de serviços originalmente contabilizados no portal.

O segundo tópico de observação traçou o perfil de acesso aos serviços governamentais brasileiros. A ideia foi a de identificar a quantidade de serviços "online", ou seja, os que podem ser consumidos pela Internet através de uso de computador, e os "offline", que são os que necessitam de suporte presencial, envio de documentação por correios ou até mesmo os que são prestados por telefone. O resultado, detalhado no Gráfico 2, demonstra que, apesar de listados no portal, 40% dos serviços governamentais brasileiros ainda podem ser considerados como "offline".

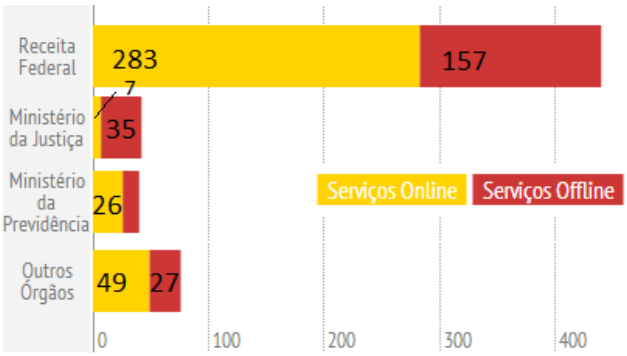


Gráfico 2. Principais provedores do e-gov brasileiro

O próximo ponto foi a análise dos métodos de autenticação dos usuários dos serviços online. Constatou-se que 75% dos serviços de e-gov transacionais disponibilizados pelos órgãos brasileiros adotam o modelo tradicional, ou em silo, onde cada provedor de serviço tem sua própria base de usuários, que se autenticam através da utilização do par usuário e senha. Percebe-se, entretanto, uma tendência pela utilização de certificado digital como forma de identificação dos usuários, já que 25% dos serviços já utilizam este método de autenticação.

Observando os sites da Receita Federal e da Previdência Social, dois dos maiores provedores de serviços do governo eletrônico brasileiro, foi possível observar como é realizado o cadastro da identidade eletrônica para serviços que não exigem a utilização do certificado digital. O processo de criação do usuário na Receita Federal se inicia com a inclusão do usuário no cadastro de pessoa física - CPF, operação que pode ser realizada tanto pelo próprio site da receita como através de instituições conveniadas. Para a geração do código de acesso aos serviços, o indivíduo precisa ter realizado alguma declaração de imposto de renda, já que a informação do número do recibo é exigida no momento da criação da senha. No caso da Previdência Social, para solicitar o cadastro da senha do usuário é necessário agendar atendimento presencial ou se utilizar da central de atendimento do órgão, que funciona através do telefone 135. Não foi possível identificar nos sites destes provedores de serviço detalhes dos modelos de autenticação, como tamanho mínimo da senha ou mesmo qual proteção utilizam contra ataques de força bruta. Ambos utilizam canal seguro (SSL) para autenticação obrigatória para utilização dos serviços.

Em relação ao certificado digital, a discussão sobre a sua utilização como principal meio de autenticação dos serviços de e-gov brasileiro não é recente, tendo sido intensificada com o lançamento do projeto piloto do Registro de Identidade Civil, no final de 2010. À época, tinha-se em mente que o cartão de identificação do cidadão viria com o certificado digital embarcado, permitindo ao brasileiro utilizar este recurso para se autenticar em sistemas governamentais e assinar digitalmente documentos. Esta ação era vista como uma tentativa de remover um dos principais obstáculos ao desenvolvimento do e-gov brasileiro, a dificuldade em se comprovar a identidade dos usuários dos serviços online.

Apesar de se caracterizar como uma tendência, os números obtidos comprovam que a certificação ainda está longe de se tornar um padrão dentro dos Ministérios e demais órgãos do governo. A constatação que corrobora esta afirmação é o fato de que 97% dos 91 serviços que utilizam certificado digital são providos por um mesmo prestador de Serviços, a Receita Federal, conforme pode ser visualizado no Gráfico 3. Além da Receita, apenas os Ministério da Justiça e do Trabalho utilizam certificação digital como prova da identidade do usuário.

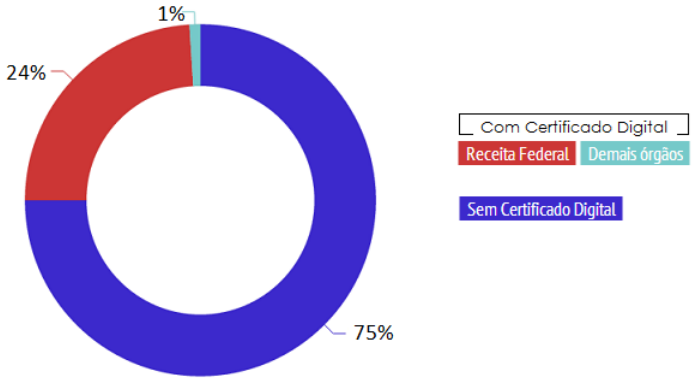


Gráfico 3. Utilização da Certificação Digital no E-gov

A última característica avaliada teve relação com os atores do processo de e-gov brasileiro. Os resultaram apontaram que aproximadamente 98% dos serviços são

direcionados ao G2C e G2B. O G2E é praticamente inexistente, apenas 1 serviço, provido pelo Ministério da Educação, foi identificado com este escopo no portal servicos.gov.br. A grande surpresa na análise deste tópico se deu pelo fato de que existem mais serviços governamentais voltados a empresas do que aos cidadãos brasileiros (Gráfico 4). Isto se dá, aparentemente, em razão de que, como já apresentado, a Receita Federal é responsável pelo provimento da maior parte dos serviços elencados, e este é um órgão que tem forte relação com o setor empresarial.

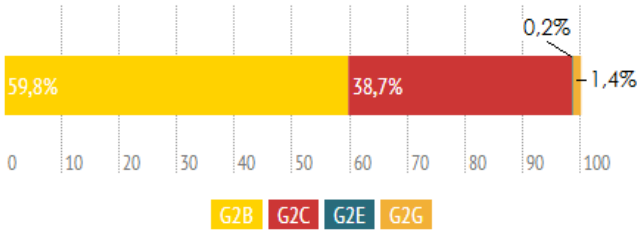


Gráfico 4. Categorização dos serviços de e-gov brasileiros.

4. Conclusão

Depois do início do processo de formalização de sua estrutura, o processo de governo eletrônico passou a ter outro ritmo de crescimento. Apesar disto, o que se percebe é que este ritmo ainda não é suficiente para que o Brasil tenha uma evolução similar à observada nos países que lideram o *ranking* de e-gov na ONU.

O que se observa, é que não há no Brasil uma estratégia nacional de Gerenciamento de Identidade, fato que tem dificultado uma evolução na maturidade dos serviços prestados pelo governo, criando uma grande diferença entre a quantidade de serviços que se encontram no primeiro e no último estágio evolutivo. A análise do portal de e-gov demonstrou, ainda, que a política de difusão do uso do certificado digital não conseguiu atingir o objetivo esperado de sanar o problema da identificação online do brasileiro nos órgãos públicos.

O diagnóstico do portal de serviços revelou, ainda, alguns fatos curiosos sobre o perfil do nosso governo eletrônico, como, por exemplo, o fato de que quase 74% dos serviços ofertados são concentrados na mão de um único provedor. Além disso, a descoberta de que mais da metade dos serviços de e-gov tem como alvo o setor de negócio fez cair por terra a ideia de que a principal motivação do governo eletrônico brasileiro é prestar um bom serviço ao cidadão.

Apesar de todas as deficiências, percebe-se uma clara evolução no desenvolvimento do e-gov brasileiro. Entretanto, para que possamos nos posicionar efetivamente dentre as potências do setor, é necessário que a política de governo eletrônico entre definitivamente na agenda de prioridades do governo federal.

5. Agradecimentos

Este trabalho conta com apoio do Ministério da Justiça, da FINEP (Projeto RENASIC/PROTO Convênio 01.12.0555.00) e da Fundação de Apoio à Pesquisa do Distrito Federal (FAP-DF).

5. Referências

Alemayehu, C., & Mwangi, J. (2011), An Interoperable Identity Management Solution for Kenya E-Government. Dissertação (Master of Science in Information Security) Departamento de Ciência da Computação, Engenharia Elétrica e Espacial, Universidade Lulea de Tecnologia, Suécia.

Bertino, E., & Takahashi, K. (2011), Identity Management: Concepts, Technologies, and Systems. Artech House.

BRASIL (2014). Diagnóstico da Identificação Civil no Brasil. Ministério da Justiça, Brasília, 2014.

DeBenedictis, A.; Howell, W.; Figueroa, R.; and Boggs, R. (2002), E-Government Defined: An Overview of the Next Big Information Technology Challenge. Issues in Information Systems, 3(1), 130-136.

Fugita, H. S., & Hirama, K. (2012). SOA–Modelagem, Análise e Design.

InfoDev, (2002), "The E-Government Handbook for Developing Countries", A project of InfoDev and the Centre for Democracy and Technology, November, Washington, D.C..

McKenzie, R., Crompton, M., & Wallis, C. (2008). Use cases for identity management in e-government. IEEE Security and Privacy, 6(2), 51-57.

OECD (2011). "Digital Identity Management for Natural Persons: Enabling Innovation and Trust in the Internet Economy - Guidance for Government Policy Makers", OECD Digital Economy Papers, No. 186, OECD Publishing, 2011.

ONU (2012). United Nations E-Government Survey: Towards a More Citizen-Centric Approach Report of the Expert Group Meeting. Disponível em http://unpan1.un.org/intradoc/groups/public/documents/un/unpan047965.pdf. Acesso em: Mar/2014.

ONU (2014). UN e-Government Survey 2014. E-Government for the Future We Want. New York:UNPAN. Disponível em http://unpan3.un.org/egovkb/Portals/egovkb/Documents/un/2014-Survey/E-Gov_Complete_Survey-2014.pdf Acesso em Jul/2014.

Shareef, M. S.; Jahankhani, H. and Dastbaz, M. (2012), "E-Government Stages: Citizen-Centric approach in Regional Government in Developing Countries," International Journal of Electronic Commerce Studies, Vol.3(1), pp.145-164.

Siau, K. and Long, Y. (2006), "Using social development lenses to understand e-government development", Journal of Global Information Management, Vol. 4 No. 1, pp. 47-62.

Takahashi, T. (Org.) (2000). Sociedade da informação no Brasil: Livro Verde. Brasília: Ministério da Ciência e Tecnologia.

Urdiales, C.; de Trazegnies, C.; Vázquez-Salceda, J.; Sandoval, F. (2004), "eGovernment and Identity Management: a Signature Coding Method for PIN generation", Presented at the EULAT'04 workshop on eGovernment and eDemocracy, Santiago de Chile.

WGID - Artigos Curtos

Armazenamento Distribuído de Dados Seguros para Efeito de Sistemas de Identificação Civil

Renata Jordão, Valério Aymoré Martins, Fábio Buiati, Flávio Elias de Deus, Rafael Timóteo de Sousa Júnior

Departamento de Engenharia Elétrica, Universidade de Brasília
Campus Universitário Darcy Ribeiro, Asa Norte, 70910-900, Brasília, DF, Brasil

{renata.jordao, fabio.buiati}@redes.unb.br,
{valeriomartins, flavioelias, desousa}@unb.br

Abstract: *In order to prevent the invasion and discovery of sensitive data, a new approach is presented to protect the confidentiality of data even when the attackers have access to all data from the server: using systems that perform queries and inserts on encrypted data without the decryption key. The model set is applied in a real scenario: a distributed system hosted by Google (EBQ - Encrypted BigQuery), which includes the use of encryption in data stored in non-relational databases for massive data processing. As a result, it´s shown that these systems support a variety of applications with low overhead.*

Resumo: *Com o intuito de impedir a invasão e a descoberta de dados sigilosos em sistemas de identificação, uma nova abordagem é apresentada para proteger a confidencialidade dos dados, mesmo quando os atacantes têm acesso aos dados do servidor, utilizando sistemas que realizam consultas em dados criptografados sem acesso à chave de decodificação. O modelo proposto é aplicado em um cenário real: um sistema distribuído hospedado pelo Google (EBQ - Encrypted BigQuery), que abarca o uso de criptografia em dados que são armazenados em bancos de dados não-relacionais (NoSQL) desenvolvidos para o tratamento massivo de dados. Os resultados indicam a possibilidade de uso do modelo em uma variedade de aplicações com baixo custo operacional.*

1 Introdução

Atualmente, o cuidado com a criação e a manutenção de ambientes seguros é uma atividade fundamental de administradores de sistemas, visto que a maioria dos ataques relatados de roubos de informações e acessos não autorizados é feita diretamente na origem dos dados (UNISYS, 2014) – seja pelo acesso aos seus bancos de dados por pessoas pertencentes à organização, ou seja pela falta de mecanismos que evitam a leitura direta das bases de dados remotamente por terceiros.

É, então, proposta deste trabalho a implementação de armazenamento distribuído de dados seguros para ambientes nos quais haja risco de acesso direto ao dado ou de uso do SGBD por outros acessos não autorizados. O objetivo desta proposta é garantir confidencialidade e integridade aos dados hospedados em bancos de dados de sistemas com conteúdo sigiloso, a partir de modelos criptográficos ajustáveis aos campos das tabelas, o que permite operações de consultas diretamente nos dados cifrados. O modelo é avaliado em um cenário real: um sistema hospedado pelo *Google EBQ*.

2. Armazenamento e processamento de dados massivos

2.1. Armazenamento de dados massivos baseados em NoSQL

O grande volume de dados gerado por aplicações *Web* tem contribuído para o surgimento de novos paradigmas e tecnologias. Nesse contexto, uma nova categoria de banco de dados, chamada NoSQL (*Not Only SQL*), foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados. Essa categoria quebra o paradigma ACID (CATTEL, 2011) e são agrupados em quatro tipos básicos, a saber:

- **Chave-valor (*Key-Value Stores*)**: modelo simples que permite a visualização do banco de dados como uma grande tabela *hash*.
- **Orientado a colunas (*BigTable-style Databases*)**: armazenador que teve como referência o modelo *BigTable* do *Google* e que, dentre suas características, podemos destacar o particionamento dos dados com forte consistência dos mesmos.
- **Orientado a documentos (*Document Databases*)**: armazena coleções de documentos, ou seja, objetos com identificadores únicos e um conjunto de campos (documentos aninhados), assemelhando-se ao modelo chave-valor; porém cada documento tem um conjunto de campos-chaves e os valores desses campos.
- **Orientado a grafos (*Graph Databases)***: possui três componentes básicos (nós, relacionamentos, propriedades) que permitem que o SGBD possa ser visto como um multigrafo rotulado e direcionado, em que cada par de nós pode ser conectado por mais de uma aresta.

2.2. Processamento de dados massivos

Hadoop foi desenvolvido para ser executado em um grande número de máquinas que não compartem memória nem discos, no qual o HDFS (*Hadoop Distributed File System*) é o sistema de arquivos distribuídos. Os dados são divididos em blocos que se propagam em diferentes servidores. Assim, os dados armazenados em um servidor – que, porventura, se desconecta ou morre – podem ser recuperados a partir de uma cópia conhecida. Para o processamento de dados, o Hadoop implementa mecanismos que executam tarefas em paralelo - o *MapReduce* - um *framework* computacional para processamento paralelo em sistemas distribuídos. São suas funções:

- *Map*: recebe uma lista como entrada; aplica uma função em que os blocos do sistema de arquivos distribuídos podem ser processados em paralelo; gera uma nova lista como saída, normalmente outros pares chave/valor.
- *Shuffle*: responsável por organizar o retorno da função *Map*, atribuindo para a entrada de cada *Reduce* todos os valores associados a uma mesma chave.
- *Reduce*: recebe o resultado da função *Map* como entrada; aplica uma função para que a entrada seja reduzida a um único valor na saída.

É alvo deste trabalho o modelo de armazenadores NoSQL orientados a colunas com funções de MapReduce implementado pela *Google*: o *BigQuery*/*Dremel*.

BigQuery é a implementação externa da ferramenta *Dremel*, usada pelo *Google* para serviço de consulta em grandes conjuntos de dados, usando uma arquitetura de processamento paralelo e orientação a colunas, o que dá ganhos significativos de E/S, pois não é lido todo o registro para cada consulta.

3. Proposta de Armazenamento de Dados Seguros

SONG et al. (2000) descrevem esquemas criptográficos para efetuar uma busca por palavras-chave em dados cifrados usando um servidor inseguro. Boneh & Waters (2007) apresentam um modelo com esquemas de chaves públicas para comparação, checagem de subconjuntos e consultas de conjunção em dados cifrados. Esses esquemas têm cifras de tamanho exponencial em relação ao texto em claro, limitando a sua aplicação prática.

Das implementações existentes, o *CryptDB* (POPA et al., 2011) é um sistema que provê confidencialidade para aplicações baseadas em bancos de dados SQL, utilizando um *proxy* para a interceptação das consultas vindas do cliente, cifrando toda a comunicação. Boneh et al. (2013) apresentam o *Cipherbase:* um sistema de banco de dados SQL que permite às organizações utilizar as vantagens da computação em nuvem mantendo a confidencialidade dos dados sensíveis.

Neste trabalho emprega-se a implementação de segurança em modelos de armazenamento orientado a colunas *BigQuery* da Google, que provê uma interface de acesso criptografado, o EBQ - uma interface que permite a cifragem dos dados, por parte do cliente, seguida do seu carregamento (*upload*) (TIGANI; NAIDU, 2014). O EBQ inclui um campo extra (*encrypt*) ao armazenamento tradicional, que determina os tipos de esquemas criptográficos que serão usados em cada coluna. Todo o acesso ao dado armazenado se dá de forma distribuída pela integração da solução com o *framework Hadoop* no *Google BigQuery* junto à plataforma de armazenamento (Figura 1).

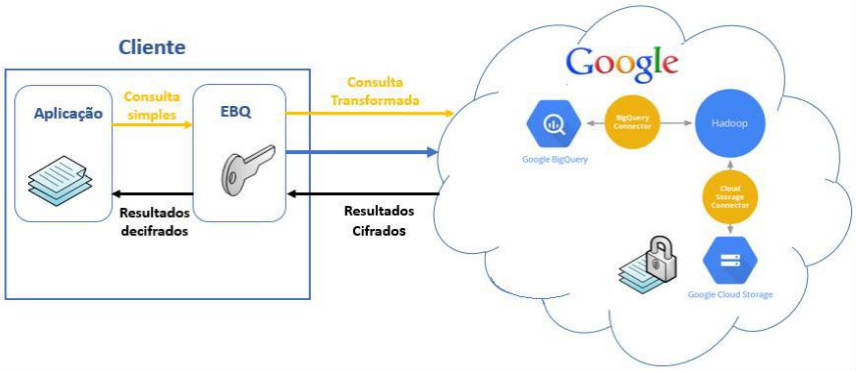


Figura 1. Arquitetura EBQ.

Assim, a consulta de dados criptografados utiliza, para cada tipo de criptografia definido, um conjunto de criptossistemas que trate esses dados adequadamente, sem a perda de confidencialidade (Goldreich, 2009). São esses:

- **Algoritmo probabilístico:** a cifra é dita probabilística quando para valores diferentes existirem cifras diferentes com grande probabilidade (KAUFMAN et al., 2002). Apesar de garantir confidencialidade e integridade, não é possível realizar manipulações com os dados cifrados, exceto o comando *SELECT*.
- **Algoritmo determinístico:** O modelo de criptografia é dito determinístico quando é gerada a mesma cifra para a mesma mensagem em claro. Esse esquema permite a realização de consultas de igualdade, isto é, pode realizar o comando *SELECT* com predicados de igualdade, junções de igualdade, *COUNT* (POPA et al., 2011).
- **Busca por palavras:** nesse esquema é calculada a função *hash* de todas as sequências possíveis de palavras. Em seguida, os *hashes* são mantidos em um campo

e separados por espaços; e asim pode ser usada uma cláusula *WHERE* com checagem de conteúdo (*CONTAINS*) com palavras-chaves inteiras, mas não aparece em consultas *SELECT* simples.

- **Busca probabilística por palavras:** consiste em buscar uma palavra com o retorno de todas as posições em que ela aparece no texto em claro (SONG et. al, 2000). Permite consultas com o uso da cláusula *WHERE* e atributo *CONTAINS* ou *LIKE*.
- **Criptografia homomórfica:** criptografia que permite lidar com tipos específicos de cálculos em cifras e gerar um resultado também cifrado que, quando decifrado, corresponde ao resultado de operações realizadas em textos em claro (Gentry, 2009).

A Tabela 1 apresenta um resumo desses modelos.

Tabela 1. Resumo dos Esquemas Criptográficos Existentes no EBQ.

Esquema	Construção	Funções	Sintaxe SQL
Probabilístico	AES em CBC	Dado estático	SELECT, UPDATE, DELETE
Homomórfico	Paillier (1999)	Adição	SUM, +
Busca por palavras	Hash das palavras	Busca de Palavras	SELECT - WHERE, CONTAINS
Busca probabilística por palavras	Song et al. (2000)	Busca de Palavras	SELECT - WHERE, CONTAINS, ILIKE
Determinísticos	AES em CBC com VI nulo	Igualdade	=; !=; IN; COUNT

4. Análises e Resultados

Foi aplicado modelos criptográficos que atendessem todos os tipos criptográficos em estudo, avaliando o desempenho e o custo operacional adicionados por cada criptossistema quando aplicado em um sistema distribuído de armazenamento massivo de dados orientado a colunas suportado do EBQ sobre o *Google BigQuery*.

Para análises de desempenho e *overhead* foi utilizado um ambiente com um computador com processador core I5 1,8GHz, 6GB de RAM e *Linux Ubuntu* 14.04; executando a ferramenta EBQ conectada ao *Google BigQuery*. O EBQ foi executado por meio de uma ferramenta baseada em *Python,* que acessa *BigQuery* usando a linha de comando.

Foi utilizado um esquema orientado a colunas, tendo como tema um cadastro pessoal que possui um número de identificação e diversas famílias de colunas, aonde um tipo de criptografia foi escolhido para cada coluna de forma a permitir a análise de sua influência no desempenho da consulta. Foi realizada a inserção com dois volumes de entrada (5000 e 50000 registros) para a análise do impacto do volume de dados criptografados em relação ao tempo de resposta.

4.1. Realização das Consultas

Para suporte a análise foi realizada uma série consultas SQL com atributos e resultados diversos tendo em vista a avaliação da viabilidade da aplicação dos criptossistemas em grandes volumes de dados. Destacam-se: (a) Consultas simples sobre cada atributo de cada família de colunas; (b) Consultas com cláusula WHERE com atributo de igualdade para teste da cifra determinística; (c) Consultas com cláusula WHERE com a condição CONTAINS para teste da cifra de busca probabilística por palavras; (d) Consultas com cláusula WHERE e testes com operações algébricas para teste de cifragem homomórfica; (e) Consultas com soma de duas colunas cifradas com criptografia homomórfica.

4.2. Avaliação dos Resultados

Foi avaliado o tempo de resposta de cada consulta EBQ, usando como comparação uma tabela sem criptografia com os mesmos registros. Cada processo de consulta foi repetido 50 vezes, de forma a obter um tempo médio, uma vez que a maior parte do processamento é realizada nos servidores do *Google* em um ambiente não-controlado em que não se pode garantir a expectativa de execução correta e homogênea.

Da Figura 4, verifica-se que as consultas EBQ acrescentam um atraso ao tempo de resposta, devido ao volume de dados que os esquemas criptográficos inserem. A consulta simples apresenta valores altos, visto que requisitou uma coluna de cada família de colunas. Observa-se também que a consulta de soma homomórfica gera um tempo de resposta grande em relação à consulta sem criptografia, devido ao tempo de processamento da soma das cifras agregado ao tempo de decodificação do resultado na máquina do cliente. Mesmo que o volume de dados aumente pelo processo criptográfico, esse aumento não prejudica a eficiência nem a velocidade de consultas dinâmicas.

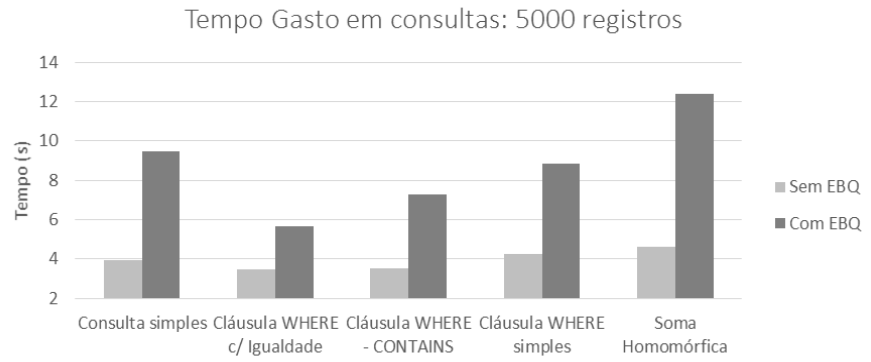


Figura 4. Tempo médio (em segundos) gasto em consultas – 5000 registros.

A mesma consulta foi realizada com volume de 50000 registros (Figura 5) aonde observa-se que, mesmo aumentando a quantidade de dados, a margem de atraso permanece aparentemente controlada, confirmando essa afirmação, apresentando a performance obtida com o número de registros aumentado em 10 vezes o seu carregamento original. Os atrasos encontrados são pequenos, se comparados com o tamanho da tabela gerada. Novamente, observa-se que as consultas mais demoradas são aquelas que envolvem a leitura de diferentes famílias de colunas e aquelas que realizam operações em cifras homomórficas, que requerem processamento extra, devido à extensão de sua cifra.

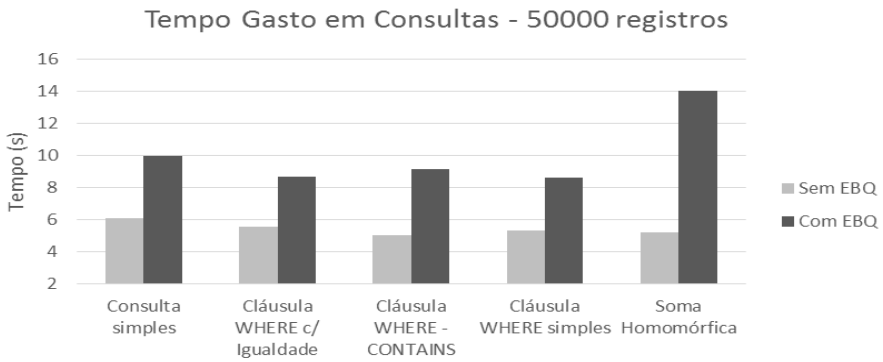


Figura 5. Tempo médio (em segundos) gasto em consultas – 50000 registros.

Verifica-se, portanto, que a aplicação de consultas diretas sobre grandes volumes de dados criptografados armazenados em ambiente distribuídos – como é o caso do EBQ – adiciona um atraso ao intervalo de resposta. Porém, à medida que o volume de dados cresce, essa variação se estabiliza.

5. Conclusões e trabalhos futuros

Os resultados apresentados permitem identificar que a utilização de criptografia em grandes volumes de dados agrega alguma carga de dados e atraso ao tempo de resposta às pesquisas, porém esse gasto extra de tempo e de memória não se torna tão relevante, levando em conta a velocidade com que as operações são efetuadas. Como proposta de trabalhos futuros é sugerida a inserção de uma gama maior de atributos de consultas, como *GROUP BY* e *DISTINCT*, assim como a realização de consultas com parâmetros de diferentes famílias de colunas. Além disso, seria interessante a implementação de um servidor seguro de gerenciamento de chaves, permitindo que a configuração de multiusuários acesse os dados criptografados.

Agradecimentos

Este trabalho conta com apoio do Ministério da Justiça, do Ministério do Planejamento, Orçamento e Gestão, da FINEP (Convênio 01.12.0555.00 RENASIC/PROTO), da Fundação de Apoio à Pesquisa do Distrito Federal (FAP-DF) e do Programa Nacional de Pós-Doutorado/CAPES in Brazil (PNPD/CAPES).

Referências

Arasu, A., Blanas, S., Eguro, K., Kaushik, R., Kossmann, D., Ramamurthy, R., & Venkatesan, R. (2013). Orthogonal Security with Cipherbase. In CIDR.

Boneh, D., & Waters, B. (2007). Conjunctive, subset, and range queries on encrypted data. In Theory of cryptography (pp. 535-554). Springer Berlin Heidelberg.

Gentry, C. (2009). A fully homomorphic encryption scheme. PhD. Stanford University.

Goldreich, O. (2009). Foundations of Cryptography: vol. 2. Cambridge University Press.

Kaufman, C., Perlman, R; Speciner, M. (2002). Network Security: Private Communication in a Public World. Editora: Prentice Hall, 2nd Edition.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residousity classes. In Advanced in cryptology – EUROCRYPT'99. Springer Berlin Heidelberg.

Popa, R. A., Redfield, C., Zeldovich, N., & Balakrishnan, H. (2011). Cryptdb: protecting confidentiality with encrypted query processing. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (pp. 85-100). ACM.

Song, D. X., Wagner, D., & Perrig, A. (2000). Practical techniques for searches on encrypted data. In Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on (pp. 44-55). IEEE.

Tigani, J., & Naidu, S. (2014). Google BigQuery Analytics. Editora: John Wiley & Sons. 1st Edition.

Unisys. (2014). Critical Infrastructure: Security Preparedness and Maturity. Independently conducted by Ponemon Institute LLC.

Um Estudo Sobre Autenticação Federada no Acesso a Recursos Computacionais por Terminal Remoto Seguro

Marcelo M. Galheigo, Antônio Tadeu A. Gomes
Laboratório Nacional de Computação Científica (LNCC/MCTI)
Av. Getúlio Vargas, 333 – Quitandinha, Petrópolis/RJ – 25651-075
{galheigo,atagomes}@lncc.br

Resumo. *O primeiro objetivo deste artigo curto é apresentar os resultados preliminares de nosso estudo de métodos para autenticação federada em recursos computacionais por terminal remoto seguro usando o protocolo SSH. Esses resultados apontam para a inexistência de uma solução que mantenha a interface de acesso típica dos terminais remotos seguros e não leve à reformulação ou reconfiguração dos provedores de identidade de uma federação. Nesse sentido, outro objetivo deste trabalho é apresentar o desenho de uma solução com as características supracitadas. Planejamos empregar essa solução no acesso remoto por linha de comando na grade computacional do Sistema Nacional de Processamento de Alto Desempenho – SINAPAD. Vislumbramos, contudo, que a solução possa ser adaptada em outros contextos.*

1. Introdução

Federações de identidade permitem que a identidade eletrônica de indivíduos possa ser armazenada e utilizada entre sistemas de gestão de identidade distintos [Shim et al., 2005]. No contexto da Internet, diversas soluções de federação de identidade têm sido desenvolvidas e adotadas. Em geral essas soluções têm como base as tecnologias de web. Dentre elas, podemos citar a linguagem SAML [SAMLCoreV2, 2005], adotada na solução Shibboleth[1], e os protocolos OAuth [RFC6749, 2012] e OpenID [OpenIDCoreV1, 2014], adotados em serviços como Google, Twitter e LinkedIn. Um elemento fundamental dessas soluções é a funcionalidade de *Single Sign-On* – SSO. Apesar do amplo uso dessa funcionalidade em aplicações web, o mesmo não acontece com aplicações "não-web". Um exemplo desse tipo de aplicação, de particular interesse para este trabalho, é o de acesso a recursos computacionais por terminal remoto seguro. Essa modalidade de acesso é empregada em grande parte na administração de recursos computacionais remotos, mas também tem seu uso difundido entre usuários de recursos especializados, como os providos por grades e *clusters* computacionais para a execução de aplicações de processamento de alto desempenho (HPC – *High-Performance Computing*). A solução para essa forma de acesso mais amplamente empregada nos dias de hoje é provavelmente por intermédio do protocolo SSH [RFC4251, 2006].

O primeiro objetivo deste artigo é apresentar os resultados preliminares de nosso estudo de soluções de autenticação federada para acesso SSH. Esses resultados apontam para a inexistência de uma solução que mantenha a interface de acesso típica dos terminais remotos seguros e não impacte na reformulação ou reconfiguração dos provedores de identidade (IdPs – *Identity Providers*) de uma federação. Nesse sentido, outro objetivo deste trabalho é apresentar uma solução concebida em nosso projeto *SSH*

[1] http://www.shibboleth.net

Federated remote authentication – SFera –, que faz parte do Programa de Gestão de Identidade da RNP. A característica chave dessa solução é a de tentar concentrar as adaptações necessárias, o quanto possível, nos provedores de serviço (SP – *Service Providers*). O raciocínio no qual essa solução se baseia é o de que SPs são os principais interessados em facilitar o acesso dos usuários aos seus serviços.

O restante deste artigo está organizado como se segue. Na Seção 2, levantamos um conjunto básico de requisitos que julgamos importantes para uma solução de autenticação federada no acesso remoto por terminal seguro. Na Seção 3, apresentamos as soluções estudadas e discorremos sobre suas vantagens e desvantagens. Na Seção 4, apresentamos em linhas gerais a solução sendo concebida como parte de nosso projeto SFera. Por fim, na Seção 5 apresentamos nossas considerações finais.

2. Requisitos

Em nosso estudo, consideramos os seguintes requisitos para a adoção de soluções de autenticação federada para acesso SSH tanto pelos usuários como pelos IdPs e SPs:

- A solução precisa ser intuitiva para o usuário. Idealmente, o usuário deveria informar apenas dados referentes a sua autenticação federada na própria tentativa de acesso SSH (seja em linha de comando ou com uma aplicação como PuTTY[2]);
- A solução não pode impactar na reformulação/reconfiguração de IdPs e demais componentes de gestão de identidades de uma federação, uma vez que isso implicaria em mudanças que afetariam a federação como um todo. Isso envolve também manter a liberdade de cada IdP implementar a sua interface de autenticação;
- Como decorrência do requisito anterior, as funcionalidades adicionais necessárias à solução devem ser de inteira responsabilidade dos SPs;
- O cliente SSH deve poder ser configurável para acionar módulos de verificação adicionais que dependam das características de um SP particular. Um exemplo dessa necessidade é o da obrigatoriedade de um usuário de uma grade computacional de preencher eletronicamente um termo de compromisso de bom uso dos recursos dessa grade, antes do acesso efetivo a esses recursos.

3. Estudo das soluções existentes

Nosso estudo abordou as seguintes soluções: a tecnologia Moonshot,[3] cujo desenvolvimento é liderado pela rede acadêmica britânica Ja.net; os Serviços para Transposição de Credenciais de Autenticação Federadas – STCFed[4] – desenvolvidos pela UFSC em parceria com a Rede Nacional de Ensino e Pesquisa – RNP; e as extensões à solução Shibboleth para autenticação não-web.[5]

3.1 Moonshot

Moonshot é uma tecnologia que emprega o protocolo RADIUS/RADSEC [RFC6614, 2012], a API GSS e o protocolo EAP [RFC7055, 2013] como base para autenticação federada. Essa mesma arquitetura, à exceção da API GSS, é utilizada no serviço Eduroam para autenticação federada em redes wifi.[6]

A infraestrutura do Moonshot contempla 3 elementos principais: o Moonshot IdP/Trust Router, o Moonshot Server e o Moonshot Client. O Moonshot IdP/Trust

[2] http://www.putty.org
[3] https://www.ja.net/products-services/janet-futures/moonshot
[4] https://www.rnp.br/pd/gts2010-2011/gt_stcfed2.html
[5] https://github.com/biancini/Shibboleth-Authentication/wiki
[6] http://www.eduroam.org

Router engloba os componentes da infraestrutura que fornecem a identidade do usuário, como o servidor RADIUS, sua base de usuários (p.ex. LDAP) e sua conexão com a federação de identidades (caso exista). O Moonshot Server contém o SP Shibboleth e o serviço que se deseja disponibilizar na federação. O Moonshot Client é o cliente final propriamente dito que quer ter acesso ao serviço por meio de uma autenticação federada. Em todos os elementos da infraestrutura Moonshot é necessário o uso da API GSS para asserção dos atributos do usuário, validação das credenciais do mesmo no momento de sua autenticação, e envio dessas credenciais através do protocolo EAP.

3.1.1. Utilização para acesso SSH

Entre as várias aplicações não-web com as quais o Moonshot pode ser utilizado, está o SSH. A integração do SSH com o Moonshot se dá via servidor SSH através da API GSS. Nesse cenário, o Moonshot Server é constituído do servidor SSH e do SP Shibboleth. Para a integração, além das configurações referentes ao Moonshot Server, é necessário também a configuração do servidor SSH para utilização da API GSS (na implementação do Moonshot é usado o módulo OpenSSH). É possível ainda definir qual dos atributos SAML obtidos no momento da autenticação será utilizado como credencial no acesso SSH, através de configurações no SP Shibboleth. Feito isso, para receber e autorizar credenciais GSS-EAP oriundas do Moonshot Client é preciso iniciar um servidor que ofereça a API GSS. No Moonshot Client, apenas as configurações referentes ao Moonshot são necessárias. Feitas essas configurações, basta que o usuário informe sua credencial de usuário federado e as envie ao Moonshot Server através de uma ferramenta cliente da API GSS.

Após esses procedimentos, o servidor Moonshot/OpenSSH/Shibboleth está pronto para receber acessos SSH do cliente que enviou sua credencial GSS-EAP através da ferramenta cliente da API GSS.

3.1.2. Vantagens e desvantagens

Assumindo que todas as instituições da federação possuam previamente a infraestrutura RADIUS implantada (p.ex., pela utilização do Eduroam), a implantação da infraestrutura Moonshot na federação é simples. A interface com o usuário é transparente, sem a necessidade de modificações no cliente SSH, apenas o envio da credencial GSS-EAP ao servidor Moonshot via ferramenta externa. Contudo, há a necessidade de instalação dos módulos Moonshot em todas as camadas da infraestrutura, bem como configuração prévia também na máquina cliente. Além disso, as credenciais ficam expostas em um arquivo texto oculto na máquina do usuário.

3.2 STCFed

Os serviços STCFed se baseiam na emissão e validação de credenciais diferentes das aceitas pelo Shibboleth. Eles empregam o "IdP+", uma extensão da implementação de IdPs do Shibboleth acrescida de 3 serviços: o *Secure Token Service* – STS –, o *Credential Translation Service* – CTS – e o Serviço Gerador de Certificados – SGC. Além disso, para emissão de certificados o IdP+ conta com uma Autoridade Certificadora – AC – confiável para emissão de certificados, como o da infraestrutura ICPEdu da RNP[7] ou o MyProxyCA[8] em um modelo auto-confiável, por exemplo.

O STS é um serviço web implementado nos moldes da especificação WS-Trust [WS-Trust-1.3, 2012] e responsável pela ponte entre o IdP Shibboleth e a aplicação não-web. O CTS é responsável pela tradução das credenciais obtidas pela autenticação na

[7] https://www.rnp.br/servicos/icpedu.html
[8] http://grid.ncsa.illinois.edu/myproxy/ca/

federação em outros padrões, como o X.509. Essa tradução é feita obtendo as informações do usuário autenticado, através da asserção SAML de seus atributos disponíveis na federação. O SGC emite o certificado X.509 assinado pela AC confiável, utilizando os atributos SAML do usuário federado.

3.2.1. Utilização para acesso SSH

O SSH é uma das várias aplicações com os quais os serviços STCFed podem ser utilizados. A integração do SSH com esses serviços se dá via servidor SSH através do módulo GSI-OpenSSH do toolkit Globus de implementação de grades computacionais.[9] Para isso, o servidor OpenSSH deve ser devidamente configurado para utilização desse módulo e da Autoridade Certificadora – AC – que emitirá os certificados dos usuários federados.

Para o acesso SSH federado, o usuário deve acessar uma aplicação web de geração de certificados. Caso o usuário não esteja ainda autenticado na federação, este será redirecionado ao site de autenticação do IdP+. Após a autenticação, a asserção SAML é utilizada pelo CTS para fazer a tradução das informações do usuário na requisição do certificado a ser enviada pelo SGC à AC confiável. Uma vez que a AC emita o certificado do usuário, este é devolvido à aplicação web, que o disponibiliza para download pelo usuário. De posse do certificado assinado pela AC, o usuário poderá acessar o servidor SSH utilizando a autenticação com base em certificado X.509.

3.2.2. Vantagens e desvantagens

A solução com STCFed (usando o IdP+) não exige a instalação de ferramentas adicionais na máquina cliente. Uma vez que o usuário possua o certificado e o servidor SSH esteja configurado para aceitar aquele certificado, o acesso SSH é transparente para o usuário. Contudo, independente da forma como a emissão de certificados pela(s) CA(s) é gerida (seja ela centralizada pelo administrador da federação ou distribuída entre os IdPs), o IdP+ traz para a camada que provê a identidade a responsabilidade de prover o serviço de gerência e emissão de credenciais.

3.3 Extensões à solução Shibboleth para autenticação não-web

Essa abordagem envolve uma série de tecnologias, das quais as principais são: a extensão *Enhanced Client or Proxy* – ECP – do IdP Shibooleth;[10] a biblioteca libcurl para manipulação programática de requisições e respostas a sites web;[11] e o serviço *Name Service Switch* – NSS – integrado à interface *Pluggable Authentication Modules* – PAM – para consulta de informações sobre usuários e tratamento da autenticação dos mesmos em ambiente Unix/Linux[12].

3.3.1. Utilização para acesso SSH

Neste modelo, para que o acesso SSH esteja disponível aos usuários federados, é necessária algumas configurações no IdP, no SP e no servidor SSH. Primeiramente devem ser feitas algumas configurações no IdP Shibboleth para habilitar o módulo ECP e para a utilização do método de autenticação HTTP Básica. Isso se deve ao fato de a integração NSS-PAM só permitir esse tipo de autenticação para consumir e inserir valores nas variáveis de login (username e password) do site de autenticação do IdP Shibboleth através da utilização da biblioteca libcurl.

---

9 http://dev.globus.org/wiki/GSI-OpenSSH

10 https://wiki.shibboleth.net/confluence/display/SHIB2/ECP

11 http://curl.haxx.se/libcurl/

12 https://amd.co.at/adminwiki/PAM/NSS

No provedor de serviço SP Shibboleth é feita a configuração para informar que a autenticação básica no IdP também será aceita. Após isso é configurada a integração NSS-PAM no servidor SSH do SP. Após essas configurações, o servidor SSH do SP já está apto a receber autenticações federadas.

3.3.2. Vantagens e desvantagens

Essa abordagem não exige a instalação de ferramentas na máquina cliente. Além disso, uma vez que o servidor SSH esteja configurado para integração com o IdP Shibboleth via NSS-PAM, o acesso SSH é transparente para o usuário. Contudo, ela envolve uma série de configurações a serem feitas na camada do IdP Shibboleth, o que pode trazer dificuldades em sua operacionalização na federação. A exigência de os sites de autenticação dos IdPs Shibboleth serem padronizados para o método de autenticação HTTP Básica é um fator limitante a flexibilidade natural inerente ao contexto de federação, onde em princípio cada IdP teria liberdade de implementar seu método de autenticação. Há ainda o fato de esta abordagem não suportar o uso do serviço *Where Are You From* - WAYF – pelo usuário federado, sendo este limitado a um único IdP acessível pelo servidor SSH.

3.4 Análise geral das soluções

Foi possível identificar que cada uma das soluções analisadas acima é aplicável no acesso SSH a recursos computacionais. Porém, nenhuma delas atende plenamente todos os requisitos apresentados na Seção 2.

4. O Projeto SFera

O projeto SFera visa fornecer mecanismos de acesso SSH federado de forma a atender os requisitos apresentados na Seção 2. Seu objetivo é prover uma interface o mais transparente possível para o usuário final, eximindo os IdPs da responsabilidade de prover e administrar tal serviço. Para alcançar esse objetivo, desenhamos uma solução, ilustrada na Figura 1, cuja implementação pode ser concentrada totalmente no SP que desejar ofertar o serviço de acesso SSH federado. Essa solução prevê ainda que o usuário possa escolher qual é o seu IdP de origem através do serviço WAYF. Há também uma maior flexibilidade nesta solução através do conceito de integração de módulos de verificação de terceiros.

Na implementação desta solução planejamos usar o módulo ECP do IdP Shibboleth e a biblioteca libcurl, bem como implementar um plug-in para o servidor OpenSSH de modo que esse possa estender as funcionalidades de seleção do IdP de origem a partir do serviço WAYF bem como ser acoplado a módulos de verificação de terceiros. Uma vez que essa abordagem permita que o usuário selecione seu IdP de origem, uma dificuldade desta solução é garantir a flexibilidade/heterogeneidade do site de autenticação de cada IdP da federação. Para isso é preciso implementar *wrappers* que permitam a injeção das credenciais do usuário para o site de autenticação do IdP de origem de forma customizada. Esses *wrappers* podem usar diferentes tipos de credenciais – como *username*/*password*, CPF/*password*, e-mail/*password*, entre outras – de acordo com o exigido pela instituição provedora da identidade. Porém, essa complexidade pode ser encapsulada no provedor de serviços, que é o principal interessado em facilitar o acesso dos usuários.

5. Considerações finais

O problema da identificação de com qual IdP um usuário está associado é particularmente complexo em aplicações não-web, como no acesso SSH. Essa

complexidade reside no fato de que os subsistemas de autenticação dessas aplicações são em geral pobres semanticamente para lidar com a autenticação federada. No entanto, os blocos chave para adicionar essa funcionalidade estão disponíveis, como explorado nas soluções estudadas neste trabalho, sendo importante identificar os requisitos que são de fato importantes para cada tipo de aplicação. Esperamos que os passos seguintes deste trabalho – fundamentalmente, a implementação completa da solução desenhada e sua comparação com outros trabalhos, p.ex. [Wangham et al., 2012] – possam validar nossas escolhas acerca da concentração da complexidade da solução para autenticação federada de acesso SSH na implementação dos SPs.

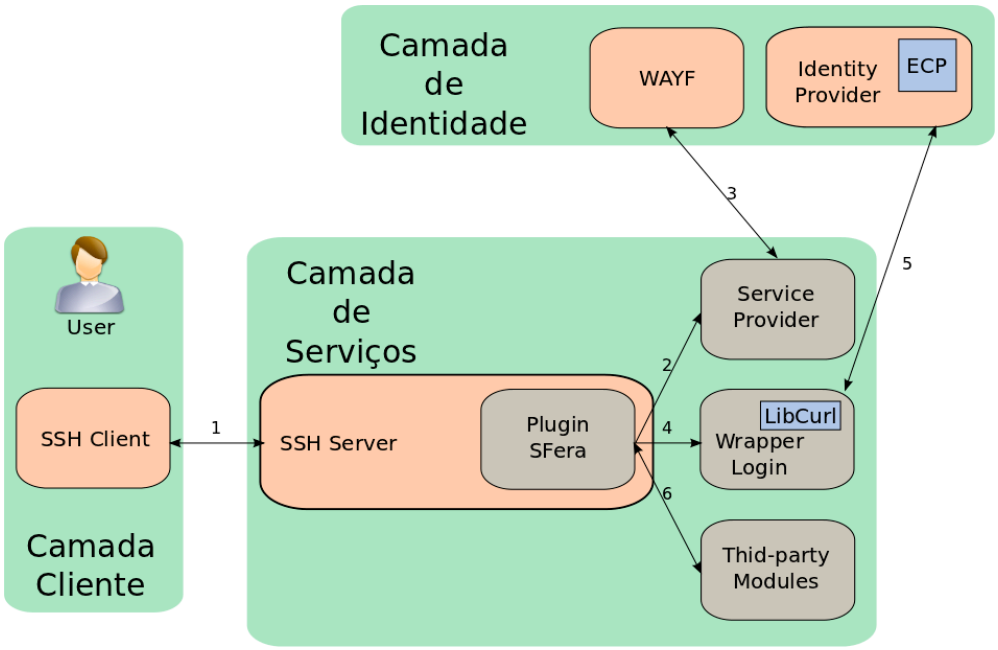


Figura 1. Arquitetura da solução SFera.

Referências

[OpenIDCoreV1, 2014] The OpenID Foundation. N. Sakimura, J. Bradley, M. Jones, B. de Medeiros e C. Mortimore (Eds.), "OpenID Connect Core 1.0," Fevereiro de 2014.

[RFC4251, 2006] IETF. T. Ylonen e C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, Janeiro de 2006.

[RFC6749, 2012] IETF. D. Hardt (Ed.), "The OAuth 2.0 Authorization Framework", RFC 6749, Outubro de 2012.

[RFC6614, 2012] IETF. S. Winter, M. McCauley, S. Venaas e K. Wierenga, "Transport Layer Security Encryption for RADIUS", RFC 6614, Maio de 2012.

[RFC7055, 2013] S. Hartman (Ed.) e J. Howlett, "A GSS-API Mechanism for the Extensible Authentication Protocol", RFC 7055, Dezembro de 2013.

[SAMLCoreV2, 2005] OASIS. S. Cantor, J. Kemp, R. Philpott, e E. Maler (Eds.), "Assertions and Protocol for the OASIS Security Assertion Markup Language V2.0", Março de 2005.

[Shim et al., 2005] S.S.Y. Shim, G. Bhalla, e V.S. Pendyala, Federated Identity Management. In: Proceedings of IEEE Computer. 2005, 120-122.

[WS-Trust-1.3, 2012] OASIS Standard. A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, e H. Granqvist (Eds.), "WS-Trust 1.3 Errata 01", 25 de abril de 2012.

[Wangham et al., 2012] M.S. Wangham, E.R. Mello, D.S. Böger, J.S. Fraga, e M.C. Guérios. Geração de Certificados Digitais a partir da Autenticação Federada Shibboleth. In: Anais do XII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg), 2012.

Um Sistema de Controle de Acesso Utilizando Agentes e Ontologia

Pedro Ricardo Oliveira[1], **João Carlos Gluz**[1]

[1]Programa Interdisciplinar de Pós Graduação Em Computação Aplicada (PIPCA)
Universidade do Vale do Rio dos Sinos (UNISINOS)
Caixa Postal 275 - 93.022-000 - São Leopoldo - RS - Brasil

pedro.oliveira@terra.com.br, jcgluz@unisinos.br

Abstract. *The increase use of learning objects and content platforms that perform the functions of storage, creation, modification and query of these objects on a managed and controlled manner, creates the need for mechanisms to control the access to these platforms. This paper aims to present a model of management and access control for a domain of a platform that supports the complete lifecycle of learning objects, using intelligent software agents and a base of ontological data.*

Resumo. *A utilização crescente de objetos de aprendizagem e plataformas de conteúdo que realizam as funções de armazenamento, criação, modificação e consulta de forma gerenciada e controlada destes objetos de aprendizagem, cria a necessidade de mecanismos de software com a função de gerência e controle de acesso a estas plataformas. O presente trabalho tem por objetivo apresentar um modelo de gerência e controle de acesso para um domínio de uma plataforma que suporta o ciclo de vida completo de objetos de aprendizagem com uso de agentes inteligentes de software e uma base de dados ontológica.*

1. Introdução

Tarefas de gerenciamento e administração são fundamentais em qualquer tipo de sistema de gerenciamento da informação. Assim, um ambiente de ensino composto por objetos de aprendizagem, usuários e aplicações que são modelados por perfis e ontologias, integrados através de agentes de software, também requer métodos e mecanismos para realização destas tarefas. A plataforma OBAA-MILOS [Vicari and Gluz 2011], que tem por objetivo criar uma infraestrutura baseada em agentes e capaz de suportar o ciclo de vida completo de um objeto de aprendizagem, de acordo com a proposta de metadados OBAA [Vicari et al. 2010], é constituída por vários sistemas multi-agente capazes de auxiliarem nas atividades de autoria, busca, uso e gerência de objetos de aprendizagem.

Inserido no projeto OBAA-MILOS, este trabalho propõe um modelo de autenticação e autorização de usuários e agentes para a infraestrutura MILOS [Gluz 2010], fazendo uso das tecnologias já utilizadas, como agentes inteligentes e ontologias. O resultado esperado será um modelo e uma arquitetura capaz de suportar o controle de acesso aos recursos da plataforma, levando em consideração todas as interfaces, agentes (usuários e aplicações) e recursos.

2. Trabalhos Relacionados

Trabalhos e ferramentas relacionados ao tema foram analisados onde destacamos que o uso de bases ontológicas já representa um importante tema de pesquisa, com resultados positivos em diversas áreas. Ferramentas para controle de acesso tradicionais, como Spring Security [Spring Security 2014] e OpenLDAP [OpenLDAP 2014] foram estudadas e observou-se que não apresentaram um grau satisfatório de integração com o ambiente da infraestrutura MILOS. Ainda, o mecanismo de autorização com e sem o uso de sessão, para ambas ferramentas teria de ser desenvolvido tal qual apresentado neste trabalho. Estes fatores foram analisados e decidiu-se descartar tais ferramentas.

Dentre os trabalhos selecionados, Onto-ACM [Choi et al. 2014] apresenta um modelo de análise semântica que propõe resolver as questões que envolvem controle de acesso em ambientes de computação em nuvem, onde prestadores de serviços e usuários precisam ser tratados de forma diferente em relação ao acesso às informações, sem causar um grande impacto ou complexidade na configuração de seus papéis. O resultado é um modelo de controle de acesso sensível ao contexto, baseado em processamento de ontologia e método de análise semântica.

Em [Katal et al. 2013] encontramos um modelo de autenticação e autorização baseado em ontologia. O trabalho implementa RBAC (Role-Based Access Control) em um domínio específico, neste caso uma universidade, utilizando uma ontologia para representação do acesso. Os papéis são apresentados em forma de classes da ontologia com permissões associadas a estas classes. O acesso é realizado em duas etapas, autenticação e autorização.

O estudo destes trabalhos permitiu observar que a criação de uma ontologia de acesso utilizando um modelo RBAC, armazenada em base local ou distribuída, utilizando a base ontológica já existente na infraestrutura MILOS, em conjunto com a implementação de algoritmos para controle de acesso utilizando agentes de software, representa a estratégia ideal para atingir os objetivos definidos neste trabalho.

3. A Infraestrutura OBAA-MILOS

O padrão de metadados OBAA proposto por [Vicari et al. 2010], tem como objetivo fornecer mecanismos de interoperabilidade de objetos de aprendizagem em plataformas heterogêneas, como: TV Digital, dispositivos móveis e Web. Este objetivo é atingido através do uso de tecnologias de agentes, sistemas multi-agente, objetos de aprendizagem e computação ubíqua, que permitem a especificação de padrões para OA (Objetos de Aprendizagem), possibilitando a autoria, o armazenamento e a recuperação destes objetos. A infraestrutura MILOS (Multiagent Infraestrutucture for Learning Object Support) [Gluz and Vicari 2010] é constituída por um conjunto de tecnologias que oferecem suporte aos requisitos do padrão OBAA.

4. Modelos de Controle de Acesso

Composto por uma etapa de autenticação e outra de autorização, de acordo com [Ramachandran 2002], um modelo de controle de acesso deve fornecer uma referência de alto nível, independente de domínio e de implementação para a arquitetura e projeto de mecanismos de acesso.

O modelo de controle de acesso baseado em papéis ou funções (RBAC), é o modelo dominante tanto na pesquisa acadêmica como em produtos comerciais. Teve sua aceitação generalizada devido à sua arquitetura simplificar a administração de segurança, incluindo herança semântica de papéis e permitindo uma definição de política de segurança abrangente, com fácil revisão das atribuições agente-papel e papel-permissão [Ferraiolo et al. 2003].

5. Modelo para Gerência e Controle de Acesso à Infraestrutura MILOS

Segundo [Wooldridge 2008], um sistema multi-agente é formado por dois ou mais agentes, interagindo entre si através de comunicação. A MILOS representa uma infraestrutura típica de um sistema multi-agente. Nestes sistemas, características como dados e controle distribuídos, diversidade de conhecimento, objetivo global decomposto em objetivos a ser atingidos por cada agente, multiplicidade de funções e um certo grau de autonomia devem ser encontradas.

O modelo para gerência e controle de acesso utilizado neste trabalho é o modelo RBAC. A tarefa de pesquisa e análise dos trabalhos relacionados e suas contribuições, permitiu selecionar as tecnologias mais adequadas e compatíveis para integração à MILOS. O modelo oferece gerência de usuários, recursos e perfis de usuários, autenticação de fator único e informações de autorização modeladas em uma ontologia. De acordo com as diretrizes da infraestrutura MILOS, o modelo apresenta a arquitetura em camadas onde um ou mais agentes implementam as funcionalidades definidas. Estas camadas são: Interface, manutenção e controle de acesso, agentes de acesso às informações e a camada de base de conhecimento.

Na camada de manutenção e controle de acesso estão os agentes responsáveis pelas operações de inclusão, alteração e exclusão das informações de acesso. Também estão nesta camada os agentes de controle de acesso responsáveis pela autenticação e autorização. Para acesso aos dados do modelo, os agentes de manutenção e controle de acesso fazem uso dos serviços oferecidos pelos agentes *Query* e *Update*, que compõem a camada de acesso às informações. A base de conhecimento é modelada por uma ontologia, apresentada na Figura 1.

A classe *Access* realiza o relacionamento entre um agente usuário *(Agent)*, um recurso e um modo de acesso permitido. O agente usuário *(Agent)* poderá ser um *User* ou uma *Application*, um recurso poderá ser um *Learning Object* ou uma *Application As Resource* quando desejarmos tratar uma aplicação como um objeto que poderá ou não ser executado. Os modos de acesso foram definidos como *Control, Read, Write, Delete e Execute*.

A base de dados ontológica [da Silva and Gluz 2012] da infraestrutura MILOS, suportada por uma camada TDB de JENA [JENA 2013] foi estendida de forma a armazenar também os elementos da ontologia de acesso. Esta base de dados TDB poderá ser distribuída em um ou mais servidores, de acordo com a necessidade da plataforma.

O modelo suporta solicitações de acesso dentro de uma sessão (exigindo um prévio login) ou sem necessidade de sessão. Solicitações sem sessão serão utilizadas por serviços externos ao ambiente e que estejam previstos na ontologia de acesso. Estas solicitações realizam a autenticação e autorização em um único passo.

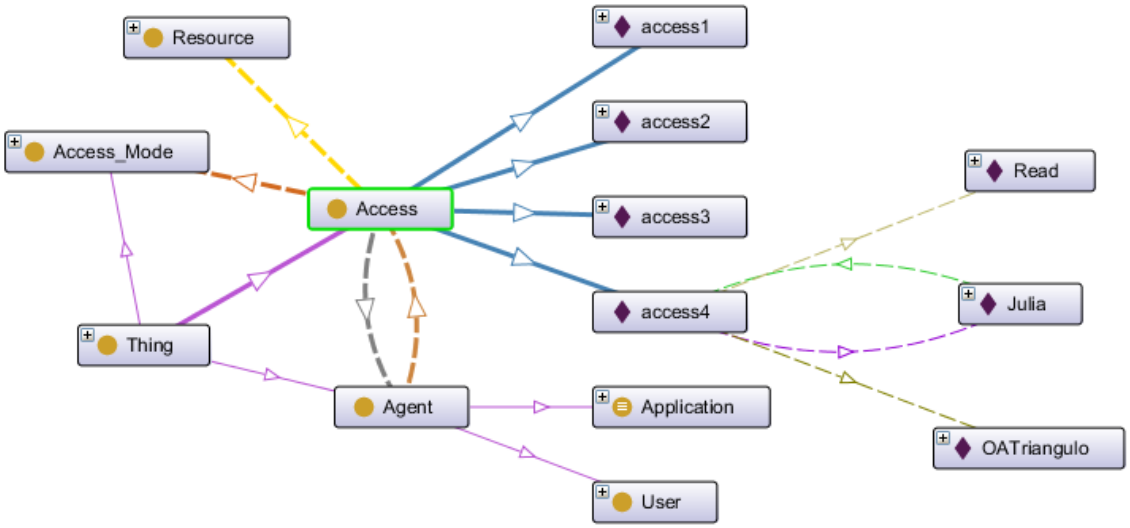


Figura 1. Ontologia de Acesso

O modelo suporta ainda controle de acesso federado, isto é, caso um agente não consiga validar o acesso por não localizar o usuário solicitante na sua base de dados, reenvia a solicitação aos demais agentes de controle de acesso do ambiente, a fim de que todas as bases distribuídas sejam pesquisadas para a validação do acesso. Mensagens de configuração permitem especificar ao agente seus parâmetros de funcionamento e uma mensagem de logout encerra uma sessão. O modelo possui um mecanismo de logout automático por inatividade em um espaço de tempo configurável.

6. Experimentos e Validações

Como metodologia para realização dos experimentos e suas validações, foram criados cenários de configurações (usuários, perfis e recursos) e casos de uso (ações de acesso), verificando se o modelo apresentava o resultado esperado. Tanto os experimentos de acesso com e sem uso de sessão, quanto o acesso federado, obtiveram resultados positivos, indicando a implementação correta do modelo.

Também foram realizados testes de desempenho, com o objetivo de medir o tempo médio de resposta do modelo, procurando dimensionar o impacto da sua implementação no ambiente atualmente em uso da infraestrutura MILOS.

7. Medidas de Desempenho

Com o objetivo de observar a escalabilidade do modelo, foram criados e executados lotes de requisições, medindo-se o seu tempo de execução. Foram utilizados lotes de 500, 1000, 2000, 4000 e 8000 acessos sucessivos, com respostas de ”Permitido”e ”Não Permitido”separadamente. Também foram utilizados os mesmos lotes para acessos com e sem sessão. A cada execução de lote, foi iniciado um *container* local JADE e encerrado logo em seguida, desta forma, *buffers* que eventualmente tenham sido criados, não interferiram nas execuções subsequentes. O resultado final é composto da média aritmética de três execuções de cada lote definido. A Figura 2 apresenta um gráfico comparativo destas medições, apresentando uma tendência de crescimento linear, o que indica uma boa escalabilidade do modelo.

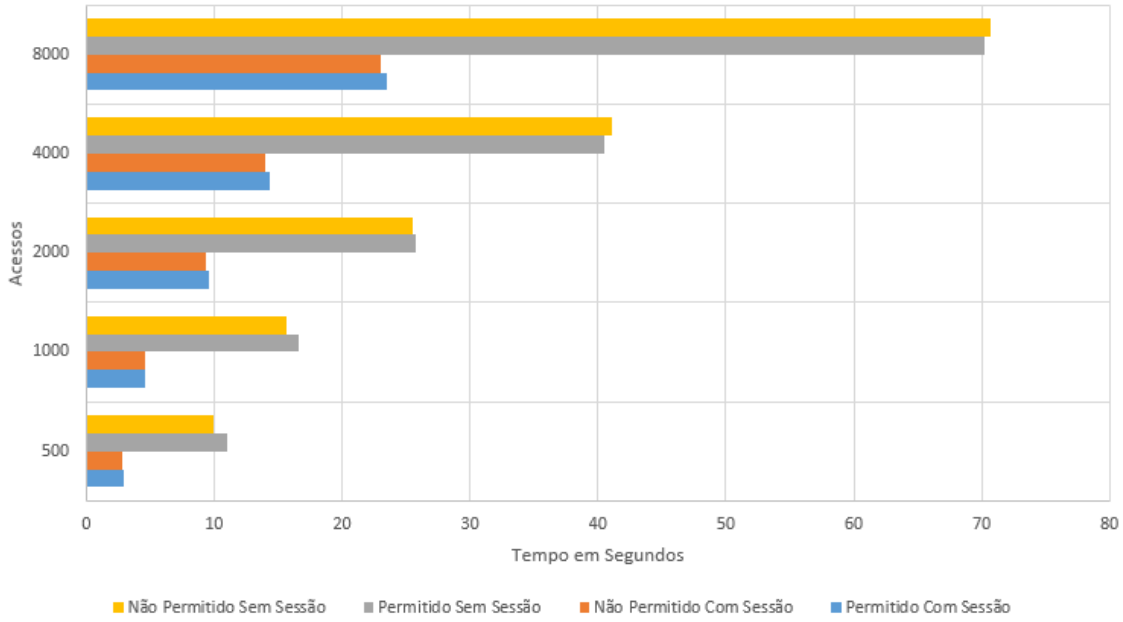


Figura 2. Tempo Médio para Execução de Acessos

Estes experimentos foram executados em uma máquina com a seguinte configuração:

- 2 Processadores Intel Xeon E5-2420 de 6 núcleos a 1.9 Gigahertz de frequência
- 32 Gigabytes de memória RAM
- Sistema Operacional Mac OS X Versão 10.9.4

8. Considerações Finais

Neste trabalho foi apresentado um modelo destinado à gerência e controle de acesso para a infraestrutura MILOS. Como o resultado final pode facilmente ser adaptado para uso em outros domínios, com ou sem a utilização de agentes, podemos dizer que se trata de um modelo abstrato, formal para controle de acesso utilizando bases ontológicas, posicionando-se em uma área de pesquisa recente, relacionada a ambientes e plataformas onde o tratamento semântico está inerentemente integrado à sua operação. Os experimentos de funcionalidade e desempenho realizados apresentaram resultados positivos e, demonstraram a viabilidade de sua aplicação.

Os trabalhos relacionados encontrados, que utilizam abordagem semântica não definem funcionalidades de autenticação e autorização em um ambiente federado, que é uma característica do modelo apresentado, no contexto de sistemas de controle de acesso baseados em ontologias e também um requisito do ambiente da infraestrutura MILOS. Com relação ao uso de agentes, os principais benefícios são a independência proporcionada, facilitando a implementação do acesso federado e a distribuição multiplataforma, uma característica do domínio onde o trabalho foi aplicado.

Como trabalho futuro, a utilização de mecanismos de coleta de dados de localização e contexto, com o processamento de inferências da ontologia de acesso, permitirá avançar o estudo do modelo tornando-o ainda mais dinâmico.

Referências

Choi, C., Choi, J., and Kim, P. (2014). Ontology-based access control model for security policy reasoning in cloud computing. *J. Supercomput.*, 67(3):711–722.

da Silva, L. R. J. and Gluz, J. C. (2012). *MSSearch: Busca Semântica de Objetos de Aprendizagem OBAA com Suporte a Alinhamento Automático de Ontologias*. Universidade do Vale do Rio dos Sinos (UNISINOS) - São Leopoldo/RS.

Ferraiolo, D., Kuhn, D., and Chandramouli, R. (2003). *Role-based Access Control*. Artech House computer security series. Artech House.

FIPA (2002). *FIPA - Communicative Act Library Specification*. Disponível em: http://www.fipa.org/specs/fipa00037/SC00037J.pdf. Acesso em: 3 jul. 2014.

Gluz, J. C. (2010). *Introdução à infraestrutura MILOS*. Disponível em: http://obaa.unisinos.br/. Acesso em: 18 out. 2013.

Gluz, J. C. and Vicari, R. M. (2010). *MILOS: Infraestrutura de Agentes para Suporte a Objetos de Aprendizagem OBAA*.

JENA (2013). *Framework for Building Semantic Web Applications*. Disponível em: http://jena.apache.org/. Acesso em: 9 dez. 2013.

Katal, A., Gupta, P., Wazid, M., Goudar, R., Mittal, A., Panwar, S., and Joshi, S. (2013). Authentication and authorization: Domain specific role based access control using ontology. In *Intelligent Systems and Control (ISCO), 2013 7th International Conference on*, pages 439–444.

Lampson, B. W. (1974). Protection. *Operating Systems Review*, pages 18–24.

OpenLDAP (2014). *Open Source Implementation of the Lightweight Directory Access Protocol*. Disponível em: http://www.openldap.org/. Acesso em: 28 jun. 2014.

Polleres, A., Gearon, P., and Passant, A. (2013). SPARQL 1.1 update. W3C recommendation, W3C. http://www.w3.org/TR/2013/REC-sparql11-update-20130321/.

Ramachandran, J. (2002). *Designing Security Architecture Solutions*. Wiley Desktop Editions Series. Wiley.

Seaborne, A. and Harris, S. (2013). SPARQL 1.1 query language. W3C recommendation, W3C. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

Spring Security (2014). *Framework for Authentication and Authorization to Java Applications*. Disponível em: http://projects.spring.io/spring-security/. Acesso em: 28 jun. 2014.

Vicari, R. M., Bez, M., da Silva, J. M. C., Ribeiro, A., Gluz, J. C., Santos, E., Primo, T., and Bordignon, A. (2010). *Proposta de Padrão de Objetos de Aprendizagem Baseados em Agentes (OBAA)*. Disponível em: http://www.portalobaa.org/padrao-obaa/artigos-publicados/. Acesso em: 24 out. 2013.

Vicari, R. M. and Gluz, J. C. (2011). *Infraestrutura OBAA-MILOS: Infraestrutura Multiagente para Suporte a Objetos de Aprendizagem OBAA*.

Wooldridge, M. (2008). *An Introduction to MultiAgent Systems*. Wiley.

Autenticação e Autorização em Federação de Nuvens

Ioram S. Sette[1], Carlos A. G. Ferraz[1,2]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
[2]Centro de Estudos e Sistemas Avançados do Recife (CESAR)

{iss,cagf}@cin.ufpe.br

Abstract. *Clouds Federation, or inter-clouds, is a trend for small and medium cloud service providers, enabling them to increase their potential capacity and, as a result, improving the quality of the cloud services offered to their customers. Nevertheless, in order to ensure privacy to user's data and services, it's necessary that cloud providers identify users and their resources in a unique way and apply the same authorisation policies and rules on them. This work proposes an authentication and authorisation model for federated clouds. This model must meet some requirements in order to grant a consistent experience for cloud federation users.*

Resumo. *Federação de nuvens é uma tendência para pequenos e médios provedores, possibilitando que estes aumentem o potencial de suas capacidades e, por consequência, melhorem a qualidade dos serviços de nuvem oferecidos a seus usuários. No entanto, para garantirem a privacidade dos dados e serviços do usuário hospedados em diferentes provedores, é necessário que estes provedores identifiquem da mesma forma os usuários seus recursos, e apliquem as mesmas regras e políticas de autorização sobre eles. Este trabalho propõe um modelo de autenticação e autorização para federações de nuvens. Este modelo deve atender vários requisitos que garantam uma experiência uniforme aos usuários da federação.*

1. Introdução e Motivação

Desde a criação do paradigma "Computação em Nuvens", o tema "Federações de Nuvens" vem ganhando relevância. Segundo Celesti *et al*. (2010a e 2010c), existirão três etapas na história da Computação em Nuvem: 1) a monolítica, onde ilhas isoladas de serviços de nuvem são providos por grandes provedores; 2) a "cadeia de fornecimento vertical", onde provedores de nuvem subcontratam serviços de outros provedores, por exemplo, em caso de transbordo; e 3) a "federação horizontal" ou "federações de nuvens" ou "internuvem", onde pequenos e médios provedores são "federados" para obterem economias de escala, uso eficiente de seus recursos, e aumentar suas capacidades.

Neste contexto, Celesti *et al*. (2010a e 2010c) definem os conceitos de "nuvem lar", provedor que provê interfaces para o usuário e contrato com o mesmo, e "nuvem estrangeira", provedor federado com a "nuvem lar" e também pode armazenar recursos do usuário sem que o mesmo tenha relacionamento com as mesmas.

O projeto CICN (Centro de Inovação em Computação em Nuvem) coordenado pelo Laboratório Nacional de Computação Científica (LNCC) propôs uma abordagem

um pouco diferente. Para obter vantagens similares às propostas pela "internuvem", a ideia foi criar o eGov, uma camada de *middleware* sobre nuvens privadas de órgãos e empresas públicas, onde os usuários das mesmas não têm conhecimento de qual nuvem utilizam. Esta camada gerencia e distribui os acessos dos usuários às federações de nuvens [Gomes 2013].

Nos dois modelos, existe a necessidade das nuvens pertencentes à federação identificarem unicamente seus usuários e recursos, bem como proverem mecanismos de autorização que apliquem políticas e regras equivalentes. Caso contrário, usuários podem experimentar comportamentos diferentes quando acessam uma ou outra nuvem.

Para garantir o acesso de uma mesma base de usuários às diversas nuvens pertencentes à federação, propõe-se que estas nuvens formem uma Federação de Identidade [Celesti *et al*. 2010a; Celesti *et al*. 2010b; Celesti *et al*. 2010c; Sanchez *et al*. 2012]. Neste modelo, os provedores de nuvem confiam nos provedores de identidade (*Identity Providers* - IdPs) para autenticar os usuários e garantir suas identidades. Como o gerenciamento de identidades é delegado aos IdPs, os usuários são identificados da mesma forma nas nuvens membros da federação [Khan *et al*. 2011; Ghazizadeh *et al*. 2012; Chadwick *et al*. 2013]. No entanto, este modelo não resolve o problema da autorização, isto é, não garante que as mesmas políticas e regras de autorização sejam aplicadas nas nuvens da federação.

"Federação de Autorização" pode ser definido como um conjunto de sistemas que possuem relação de confiança entre si e compartilham políticas e regras de autorização. Quando a Federação de Autorização também compartilha de mesma base de usuários, existe uma "Federação de Autenticação e Autorização". Chadwick *et al*. (2012) definem um serviço para autorização para provedores de nuvens baseado em ABAC (*Attribute-Based Access Control*) capaz de mesclar resultados de vários "Pontos de Decisão de Políticas" (PDP) diferentes. No entanto, este modelo não se aplica a Federações de Nuvem.

O objetivo deste trabalho é propor um modelo para Federação de Autenticação e Autorização no contexto de Federações de Nuvens.

Este artigo se divide em 4 seções. A seção 2 apresenta os requisitos necessários para uma Federação de Autenticação e Autorização em Federações de Nuvens. A seção 3 apresenta um modelo que atende aos requisitos identificados. Por fim, a seção 4 apresenta conclusões e trabalhos futuros.

2. Requisitos para Autenticação e Autorização em Federações de Nuvens

No contexto de uma Federação de Nuvens, os requisitos necessários (marcados com um asterisco) e desejáveis para uma "Federação de Autenticação e Autorização" são listados a seguir:

1. *Receber de provedores de nuvem (*Cloud Service Providers* - CSPs) requisições de autenticação delegada e de autorização com granularidade fina, de forma segura;
2. Oferecer serviços de autenticação e autorização independentes;
3. *Ser capaz de identificar usuários e prover os mesmos atributos a todos os CSPs;
4. Ser capaz de combinar atributos de identidade de múltiplas autoridades de atributos;

5. *Ser capaz de aplicar o mesmo conjunto de políticas em todos os CSPs;
6. *Garantir que o conteúdo dos dados e de serviços não serão informados a sujeitos não autorizados;
7. Não ser ponto único de falha;
8. Permitir que proprietários de dados e serviços controlem a autorização;
9. *Suporte operações de migração e replicação (detalhadas a seguir);
10. Ser extensível a controle de uso (*usage control model* - UCON) e criptografia baseada em atributos (*Attribute-Based Encryption* - ABE).

Em um contexto de Federação de Nuvens, recursos como arquivos de dados e/ou máquinas virtuais (VMs) podem ser transferidos de uma nuvem para outra por motivo de transbordo ou balanceamento de recursos. Desta forma, duas operações citadas no requisito 9 são apresentadas: migração e replicação de recursos entre nuvens.

A "migração" de recursos é uma operação onde um recurso hospedado em uma nuvem é transferido para outra nuvem. Esta operação pode ser explícita, caso o usuário ou administrador tenha conhecimento sobre as nuvens pertencentes à federação, ou implícita, realizada automaticamente pelo sistema, caso o mesmo identifique que não possui mais infraestrutura disponível para hospedar o recurso. Por recurso, podemos considerar qualquer entidade de usuário hospedada no provedor de nuvem, por exemplo, arquivos num sistema de armazenamento, ou máquinas virtuais, num sistema de computação elástica.

A "replicação" de recursos entre nuvens aumentam a garantia de disponibilidade e desempenho em seus acessos. Por exemplo, cópias de arquivos podem ser mantidas em diferentes provedores, ou instâncias de uma mesma máquina virtual podem ser executadas em diferentes provedores. Neste caso, o usuário controla o acesso a partir da nuvem "de origem" onde o recurso foi criado, e o sistema mantém o sincronismo das réplicas nas nuvens "estrangeiras".

3. Modelo para Autenticação e Autorização em Federações de Nuvens

Nesta seção, um modelo de "Federação de Autenticação e Autorização" para Federações de Nuvens é proposto. Este modelo deve atender aos requisitos definidos na seção 2.2.

O modelo, representado na Figura 1, combina uma Federação de Identidades (ou de autenticação) com uma Federação de Autorização de arquitetura distribuída e mecanismo de controle de acesso baseado em atributos (ABAC), que permite regras com granularidade fina. As conexões devem usar protocolos como SSL para garantir o sigilo dos dados trafegados. Os requisitos 1 e 2 são atendidos por estas características.

Os CSPs membros da Federação de Nuvens também participam de uma Federação de Identidade (ou de Autenticação). Portanto, a autenticação dos usuários é delegada aos IdPs federados. A integração com os IdPs são realizadas normalmente através de um módulo ou serviço de identidade (*Cloud Identity Service*), interno à arquitetura dos provedores [Sette *et al*. 2013; Sette *et al*. 2014]. Os trabalhos citados descrevem a integração de um provedor Openstack a provedores de Identidade através dos protocolos OpenID Connect e SAML. Neste modelo, o usuário deve se autenticar com seu IdP antes de acessar o CSP. Desta forma, problemas típicos da centralização, como ser um ponto único de compromisso ou de falha podem acontecer. Por exemplo, se o

IdP do usuário estiver indisponível, o serviço de nuvem não pode ser acessado. No entanto, estas desvantagens são contrapostas e justificadas por vantagens como a redução de credenciais para os usuários e a possibilidade de mecanismos de autenticação mais sofisticados. A federação de identidade garante que os usuários sejam apresentados de forma única aos provedores de nuvens federados, atendendo o requisito 3. O requisito 4 pode ser implementado pelo módulo de identidade do provedor, sendo necessário que este componente tenha como saber que os atributos pertençam a um mesmo usuário.

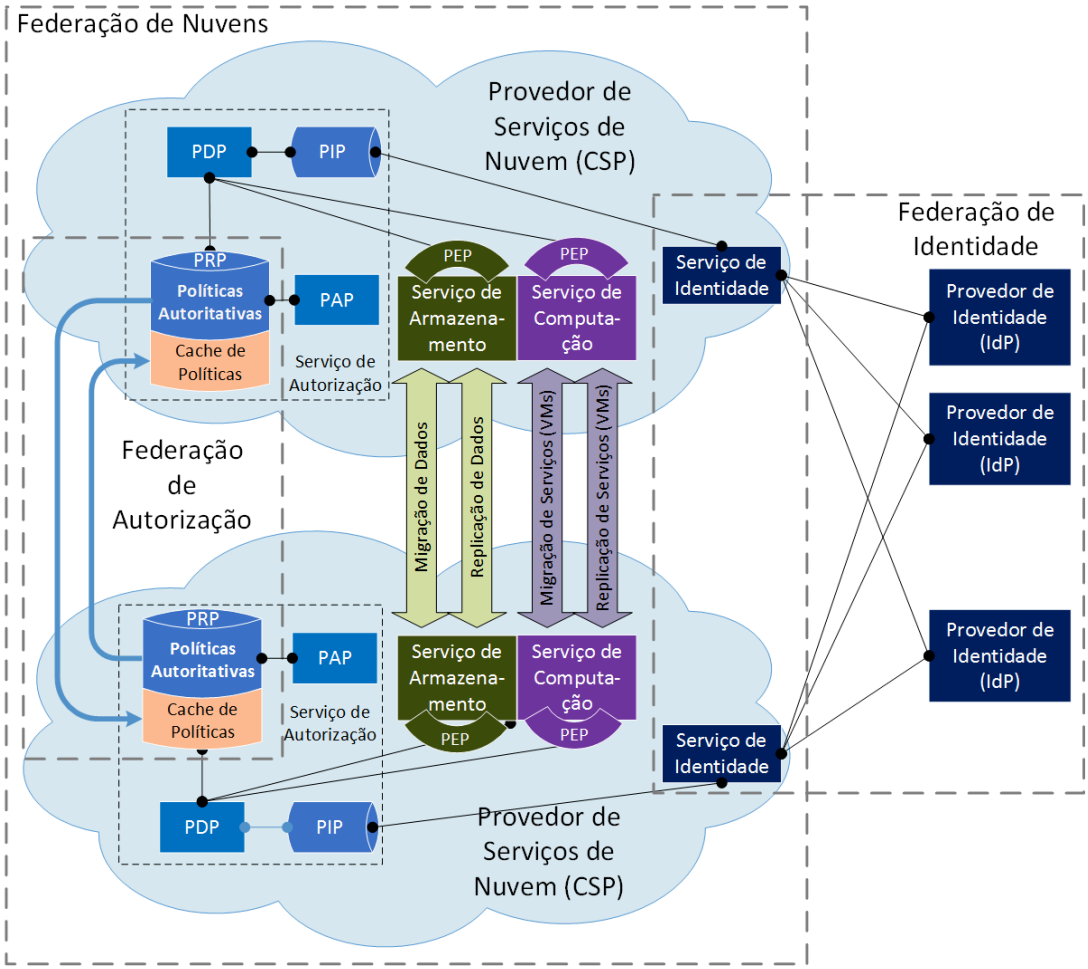


Figura 1. Modelo para Federação de Autenticação e Autorização em Federação de Nuvens

De forma análoga, os CSPs membros da Federação de Nuvens também formam uma Federação de Autorização, de forma a garantir os requisitos 5 e 6. Porém, diferente da Federação de Identidade, a arquitetura distribuída apresentada na Figura 1 permite o sincronismo de regras e políticas de acesso. Esta arquitetura foi inspirada no modelo do serviço de nomes da Internet (DNS), onde as regras de autorização são armazenadas no provedor de origem dos recursos, de forma análoga ao DNS autoritativo, e, quando a regra não está presente, ela é encontrada na rede e armazenada em *cache* no provedor local, de maneira análoga ao DNS recursivo. Desta forma, a solução ganha em desempenho, uma vez que as regras e políticas de autorização acabam, na maior parte das vezes, estando armazenadas localmente, e também em não haver um ponto único de falhas, uma vez que a solução é distribuída. O requisito 7 está, desta forma, atendido.

Cada provedor de nuvem possui um serviço de autorização, responsável pela decisão de liberação da ação solicitada pelos demais serviços (ex.: serviço de computação ou de armazenamento). O serviço de autorização utiliza um sistema ABAC composto pelos módulos definidos na arquitetura XACML (*eXtensible Access Control Markup Language*). Os serviços da nuvem que recebem requisições de acesso devem garantir que, para cada requisição, o serviço de autorização seja consultado através de um "Ponto de Aplicação de Políticas" (PEP, do inglês *Policy Enforcement Point*). O "Ponto de Decisão de Políticas" (PDP) faz interface com o PEP, recebendo a requisição de acesso e respondendo com uma permissão ou negação. Para tomar sua decisão, o PDP verifica políticas e regras de acesso armazenadas nos "Pontos de Recuperação de Políticas" (PRP). As regras e políticas sobre os recursos do usuário devem ser definidas apenas pelos usuários proprietários destes recursos, conforme o requisito 8. No entanto, regras globais podem ser definidos por administradores do projeto (*tenant*) ou do provedor de nuvem. Neste caso, o conceito de "projeto" e as "regras" precisam ser propagadas aos outros provedores federados. As regras e políticas de autorização são definidas através do "Ponto de Administração de Políticas" (PAP). Elas recebem como parâmetros atributos do sujeito (quem deseja o acesso), do objeto alvo, ou do ambiente, que são informados ao PDP pelo "Ponto de Informação de Políticas" (PIP). Os atributos do objeto são provenientes de seus metadados, enquanto os atributos do usuário são adquiridos através do módulo de identidade do CSP, provenientes dos provedores de identidade (IdPs).

Conforme previsto no requisito 9, operações de migração e replicação de dados devem ser suportadas.

Durante operações de migração de dados ou VMs, os metadados e as regras e políticas de autorização referentes ao objeto também são migrados para o novo CSP (*Cloud Service Provider*). Funciona como se o provedor de origem daquele objeto fosse alterado para o novo provedor de destino.

Já nas operações de replicação, apesar da réplica estar hospedada em um provedor diferente do provedor de origem, as regras de acesso devem permanecer no provedor de origem. Desta forma, quando um acesso é realizado a estes objetos, o PRP deve buscar na federação qual provedor tem autoridade sobre estes objetos e recuperar as regras (de acesso) referentes aos mesmos. As regras serão mantidas em um "cache" de políticas e regras por tempo definido pelo dono dos objetos. Desta forma, garante-se que as regras aplicadas às réplicas sejam as mesmas aplicadas aos objetos originais.

De acordo com o requisito 10, este modelo pode ser facilmente extensível. Para permitir um controle de uso (UCON), a verificação de acesso deve ser realizada várias vezes durante o acesso e obrigações podem ser efetuadas também durante a execução. A extensão para um mecanismo de criptografia baseada em atributos (ABE) também é possível, uma vez que utilizamos um controle de acesso baseado em atributos (ABAC). No entanto, é necessária uma reflexão sobre como trafegar chaves criptográficas entre provedores federados de forma segura, caso necessário.

4. Conclusões

Num modelo de Federação de Nuvens, usuários possuem apenas uma interface para acessar e usar seus dados e serviços, que podem estar hospedados em mais de um provedor de nuvem de forma transparente aos mesmos.

Este trabalho se propõe a definir um modelo para autenticação e autorização aplicável a Federações de Nuvens, que permita que os provedores compartilhem de uma mesma base de usuários, e que as mesmas regras e políticas de autorização sejam aplicadas por estes provedores. Objetos dos usuários podem ser migrados de um provedor para outro ou replicados entre provedores sem que o usuário perceba.

O modelo proposto é baseado em Federação de Autenticação e Autorização, com uma arquitetura de autorização distribuída baseada em ABAC. Este modelo atendeu a diversos requisitos necessários e desejáveis que foram elencados neste artigo.

Como trabalho futuro, este modelo será implementado, testado e validado em uma Federação de Nuvens formada por provedores que utilizam a plataforma Openstack.

Referências

Celesti, A. et al. (2010a) "How to Enhance Cloud Architectures to Enable Cross-Federation" In: *3rd International Conference on Cloud Computing*, p. 337-345. IEEE.

Celesti A. et al. (2010b) "Security and Cloud Computing: InterCloud Identity Management Infrastructure" In: *Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, p. 263-265. IEEE.

Celesti A. et al. (2010c) "Three-Phase Cross-Cloud Federation Model: The Cloud SSO Authentication" In: *Second International Conference on Advances in Future Internet*, p. 94-101. IEEE.

Chadwick, D. W., Casenove, M. and Siu, K. (2013) "My private cloud – granting federated access to cloud resources" In: *Journal of Cloud Computing*, p.1-16. SpringerOpen.

Chadwick, D. W. and Fatema, K. (2012) "A privacy preserving authorisation system for the cloud", In: *Journal of Computer and System Sciences*, i.78, p.1359-1373. Elsevier.

Gazizadeh, E. et al. (2012) "A trusted based model for federated identity architecture to mitigate identity theft", In: *2012 International Conference for Internet Technology And Secured Transactions*, p.376-381. IEEE.

Gomes, A. T. A. (2013) "CICN - Centro de Inovação em Computação em Nuvem", FINEP/0615/11.

Khan, R. H., Ylitalo, J. and Ahmed, A. S. (2011) "OpenID authentication as a service in OpenStack", In: *7th International Conference on Information Assurance and Security*, p.372-377. IEEE.

Sánchez, R. et al. (2012) "Enhancing Privacy and Dynamic Federation in IdM for Consumer Cloud Computing" In: *IEEE Transactions on Consumer Electronics*, vol. 58, i.1, p.95-103. IEEE.

Sette, I. S. and Ferraz, C. A. G. (2013) "Integrando Openstack com Provedores de Identidade OpenID Connect e SAML: Uma Análise Comparativa" In: *XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, p.497-506, SBC.

Sette, I. S. and Ferraz, C. A. G. (2014) "Integrando Plataformas de Nuvens a Federações de Identidade" In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p.617-630. SBC.

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

WFC – III Workshop de Forense Computacional

ZetaDnaCripto: Método de criptografia baseado em fitas de DNA.

Yasmmin C. Martins

Instituto Militar de Engenharia (IME)
Rio de Janeiro – RJ – Brasil

nim_asay@hotmail.com

Abstract. *This paper describes a method to encrypt informations using concepts of genetic strands. This method involves procedures that transform the plaintext and uses substitution cipher to compose the base sequence of ciphertext, ready to be synthesized, together with it key, molding a symmetric cipher system.*

Resumo. *Este artigo descreve um método de criptografar informações utilizando conceitos de fitas genéticas. Este método envolve procedimentos que transformam o texto em claro e usam cifra de substituição para compor a sequência de bases do criptograma, pronta para ser sintetizada, junto com sua chave, formando um sistema de cifra simétrico.*

1. Introdução

A criptografia baseada em DNA é um assunto relativamente novo, e surgiu a partir da ideia de se criar computadores genômicos. A computação em DNA é um campo da bioinformática que busca a solução de problemas através da manipulação de sequências de DNA. Como é uma área emergente, ainda estão formulando seus métodos, formalismos, como foi no início da computação convencional eletrônica [ISAIA 2004].

Hoje em dia a segurança de um sistema é essencial, não apenas para uso por governos e militares mas também em transações financeiras e comerciais. Com o avanço da tecnologia e das pesquisas, a criptografia tem ganhado métodos inovadores de cifragem. Como as curvas elípticas, a criptografia quântica, a criptografia por DNA etc.

Este artigo tem o objetivo de apresentar uma forma de criptografia baseada em fitas de DNA, com cifra simétrica, utilizando para as operações uma biblioteca chamada BioJava, construída para lidar com dados e operações relacionadas à análise genômica. Outra técnica utilizada, ao final da encriptação e no começo da decriptação, refere-se ao uso do DNA-Pascal, para fazer operações com as fitas.

Além disso, como aplicação do método proposto, será discutido o processo pelo qual poderá garantir o sigilo e a integridade de dados para os procedimentos de análise forense.

Ao longo do texto, o tipo de dado que está sendo manipulado, as bases que formam a sequência de DNA, será melhor explicado assim como algumas de suas operações; será dada também uma breve explicação sobre a biblioteca BioJava e o método DNA-Pascal;

serão discutidas outras abordagens de criptografia simétrica com DNA; então será apresentado o método proposto e por fim, sua aplicação em análise forense.

2. Computação genômica

Esta nova ciência está baseada em formas de processamento de fitas genéticas de DNA, que por sua vez, são compostas de quatro bases nitrogenadas que são adenina (A), citosina (C), timina (T) e guanina (G). As fitas de DNA possuem cadeia dupla, já as fitas de RNA, possuem uma cadeia única, com presença de outra base, a uracila (A). As combinações possíveis destas bases geram códons (grupo de três bases), estes códons podem codificar um ou mais aminoácidos diferentes, que são responsáveis por formar as proteínas.

Alguns conceitos da computação convencional podem ser abstraídas para o modelo genômico, como a representação de dados, que na primeira é feita através de 0 e 1 e na segunda por meio de 4 letras (A C T e G); na primeira as operações feitas com as informações são executadas por meio de circuitos e componentes eletrônicos, já na computação genômica, como se tratam de moléculas, que estão no núcleo das células, as operações a serem feitas com a fita de DNA, devem ser reações químicas microbiológicas.

Na formalização desta forma de computação, os autores têm abstraído operações como as da computação convencional para similar reações como síntese, hibridização, desnaturação, polimerização, técnicas de separação (para busca dentro de uma sequência, por exemplo).

As principais aplicações da computação em DNA são para aproveitar algumas das principais vantagens da manipulação das sequências, como no caso da implementação das memórias associativas que se refere à capacidade de se codificar um grande volume de informação em uma pequena quantidade de DNA (1 bit por nanômetro cúbico); o fato de poder realizar várias operações biológicas em moléculas de um mesmo tubo-teste ao mesmo tempo. Devido a este grande paralelismo, isso se torna muito vantajoso na busca de soluções para problemas combinatoriais. Em criptografia, estas propriedades se tornaram interessantes, pelo poder de esconder uma mensagem, em fragmentos pequenos que podem conter muita informação.

3. Tecnologias utilizadas

Para efetuar as etapas do método proposto foi utilizada a biblioteca BioJava por conter recursos que tornam mais simples e prática a manipulação de sequências e de seus valores de bases. Para simular as operações de síntese, concatenação e corte de sequências, no fim da encriptação e no início da decriptação, foi utilizado o método DNA-Pascal.

3.1. Biblioteca BioJava

BioJava[1] é um projeto dedicado a fornecer um framework em java que possa processar dados biológicos. Ele provê rotinas estatísticas e analíticas, parsers para leitura e escrita

[1] http://biojava.org/wiki/BioJava:CookBookLegacy

em formatos de arquivos deste domínio como fasta, GenBank, EMBL, UniProt e INSDseq.

Ele possui várias ferramentas de análise, como recursos para verificar similaridade entre sequências como o BLAST. Além de ferramentas para extrair localizações e features (características). Além de métodos para transcrição e tradução de sequências. Além do fato de se poder utilizar matriz de pesos e programação dinâmica, com estrutura também para ontologias e definição e manipulação de estruturas. Possui simulações de mutação e recombinação de acordo com uma determinada frequência que auxilia na implementação de algoritmos genéticos.

3.2. Método DNA-Pascal

De acordo com ISAIA, há 4 métodos de formalismos da computação genômica, em relação às operações e procedimentos para estas, dentre as quais estão o método de splicing, o DNA-Pascal, o de construção e o de filtragem. Foi escolhido para ser abordado neste trabalho o método DNA-Pascal, pois através de algumas operações deste método pôde-se provar que existe uma solução, em tempo polinomial, para o problema NP-Completo 3SAT [LIPTON 1994]. Enquanto os outros possuem dificuldades consideráveis de verificação e implementação.

O DNA-Pascal surgiu da união de algumas das instruções já existentes na linguagem Pascal para a computação convencional, com outras operações e testes aplicáveis à computação genômica. Apesar de o alfabeto utilizado pelas duas abordagens de computação ser diferente, o da genômica é um conjunto representado por {A, C, T, G} e o da convencional é um conjunto representado por {0, 1}, os resultados destas operações independem dos conjuntos a serem utilizados.

Este método trabalha somente com números e arrays inteiros não-negativos, também foi acrescentado um novo tipo de dado chamado multiconjunto finito de palavras. A tabela a seguir mostra as operações especiais e o que elas significam, sendo m uma variável que representa um número natural; x é uma palavra; ^ significa "elevado a", a é uma subpalavra e t, t1 e t2 são multiconjuntos finitos de palavras.

Tabela 1. Operações especiais em DNA-Pascal.

Abreviação	*Nome*	*Instrução*
(UN)	Union (União)	t := t1 U t2
(IN)	Initialization (Inicialização)	t := IN(m), IN(M) = {0,1}^m com m ≥ 0
(AS)	Assignment (Atribuição)	t := t1
(EX)	Extraction (Extração, sem retorno)	t := EX(t1, m, a), EX(t1, m, a) = t1 ∩ ({0, 1}^(m-1) a{0, 1}*), com a ∈ {0, 1} com m≥1
(SX)	Subword Extraction (Extração de palavra)	t := SX(t1, x), SX(t1, x) = t1 ∩ ({0, 1}* x{0, 1}*), com x ∈ {0, 1}*
(EW)	Empty Word (Palavra vazia)	t := {ε}
(RA)	Right Adding (Adição pelo lado direito)	t := t1 . a, t1 . a = {za | z ∈ t1}

(LA)	Left Adding (Adição pelo lado esquerdo)	t := a . t1, a . t1 = {az | z ∈ t1}
(CO)	Concatenation (Concatenação)	t := t1 . t2, t1 . t2 = {xy | x ∈ t1, y ∈ t2}
(RC)	Right Cut (Subtração pelo lado direito)	t := t1/, t1/ = {z/ | z ∈ t1}, com za/ = z para a ∈ {0, 1} e ε/ = ε
(LC)	Left Cut (Subtração pelo lado esquerdo)	t := \t1, \t1 = {\z | z ∈ t1}, com \za = z para a ∈ {0, 1} e \ε = ε
(IS)	Intersection (Interseção)	t := t1 ∩ t2

A próxima tabela possui algumas operações de teste, adicionais às operações acima.

Tabela 2. Verificações de teste auxiliares.

Abreviação	*Nome*	*Instrução*	*É verdade se:*
(SU)	Subset (subconjunto)	t1 ⊆ t2	O multiconjunto t1 contém as palavras de t2
(EM)	Emptiness (vazio)	t = ∅	O multiconjunto t é vazio
(ME)	Membership (é membro de)	x ∈ t	A palavra x pertence ao multiconjunto t

Todas estas operações são correspondentes às operações biológicas previamente apresentadas, o DNA-Pascal já foi utilizado para representar, através destas operações da tabela, o problema *Satisfability* que também é um problema NP-Completo.

4. Métodos de criptografia simétrica baseados em DNA similares

A criptografia simétrica é aquela em que a chave que codifica a mensagem é a mesma usada para a decodificação. Os algoritmos simétricos são mais rápidos que os algoritmos de chave assimétrica, onde há uma chave pública para cifrar e uma chave privada para o receptor autorizado decifrar.

Em Bhoir e Mathangi, é proposto um método bem simples de aplicar criptografia, tranformando mensagem em claro em uma sequência de bases. Este método que se baseia em substituir cada letra pelo número ASCII correspondente em formato decimal e então são agrupados em blocos e encriptados por algum algoritmo conhecido, DES por exemplo. Feito isso, o resultado é codificado para o formato binário, e cada uma das quatro bases equivale a uma combinação possível de pares 0 e 1: A para “00”, T para “01”, G para “10” e C para “11”.

Depois, são setados os primers de cada lado da mensagem, estes primers servem como indicadores de paradas e detecção da mensagem. Isso precisa ser colocado antes da comunicação ocorrer, então a sequência com a informação é adicionada depois das paradas. E por fim, pode ser confinado em microarray ou separado em muitas sequências de DNA. Para decriptar a mensagem, são feitas as operações inversas da encriptação.

Em Terec et Al, são propostos dois métodos de encriptação simétrica por DNA, e um método assimétrico também neste domínio, aqui serão falados brevemente somente os simétricos, devido ao método proposto neste artigo ser simétrico.

O primeiro método simétrico proposto foi implementado na plataforma java simples, e consiste de passos como gerar um índice aleatório, pela classe SecureRandom do pacote Java.Security, sendo este índice gerado de acordo com a limitação de 4 pois são encontradas apenas 4 opções de substituição, A, C, T e G, para posterior tradução. Depois deste passo, a chave a ser gerada deve ter o mesmo comprimento do texto em claro. No caso, este texto é a mensagem traduzida segundo o alfabeto de substituição. O tamanho da chave deve ser um múltiplo de três, então quando enviada uma mensagem longa, o comprimento da chave seria enorme. Por isso, a mensagem é separada em blocos e a cifra encripta e decripta um bloco de cada vez. A cifra de DNA então usará uma fração da mensagem original, e o único modo de implementação foi o ECB (Electronic Code Book).

ECB é o modo mais simples, em que cada bloco de texto em claro encripta um bloco de criptograma. A desvantagem deste modo é que o mesmo texto em claro sempre encriptará para o mesmo criptograma, quando se usa a mesma chave. Esta cifra por DNA comporta-se como uma segunda camada de encriptação, na qual é utilizado um HashMap, que é uma estrutura para armazenar pares chave e valor de forma única. Com isso, foi montada uma nova tabela de associação, onde as chaves são a pontuação ou as letras possíveis e o valor é um trio de formado pelas bases.

Como a tabela considera apenas letras minúsculas na associação, antes de fazer a substituição os blocos de texto em claro foram passados para letras minúsculas. E então o resultado depois da aplicação da cifra de DNA, é uma string com caracteres correspondentes ao alfabeto de DNA, e por fim, a mensagem encriptada final é um array de Bytes formado pela operação XOR entre o criptograma anterior e a chave. Para decriptar, basta fazer as operações reversas, como outro XOR entre o criptograma recebido e a chave. A quebra do que foi obtido em grupos de três, uso do HashMap com os pares valor e chave invertidos e então poderá ser lida a mensagem em claro.

O outro método simétrico proposto foi implementado utilizando a biblioteca BioJava. O primeiro passo deste método consiste em transformar cada caracter da mensagem em claro no seu valor binário segundo a tabela ASCII, após isso, uma outra função obtém destes 8 caracteres a string correspondente depois da substituição de cada par de 0s e 1s, por uma das 4 letras possíveis do alfabeto de DNA, sendo utilizado 00 para “A”, 01 para “C”, 10 para “G” e 11 para “T”.

Então, os caracteres codificados anteriormente são buscados no cromossomo escolhido como chave da sessão no início da comunicação. A mensagem em formato de DNA é separada em grupos de 4 caracteres, e cada grupo que é encontrada nas primeiras posições do cromossomo, guarda-se seu índice num vetor de localizações dos grupos no cromossomo. E usando a ferramenta de leitura de arquivos de sequências em formato FASTA, no BioJava, as sequências correspondentes aos índices em que os grupos foram achados, podem ser guardadas numa lista de objetos.

Na última fase de encriptação, para cada caracter da mensagem um índice aleatório do vetor de índices construído é escolhido. E dessa forma, mesmo que a chave seja a mesma para encriptar a mensagem, podem ser obtidos criptogramas diferentes por causa da aleatoriedade dos índices.

Na decriptação, como a chave é a mesma que a usada na encriptação, cada índice recebido com a mensagem codificada pode ser revertido para os grupos de bases, e cada base possuindo seu equivalente em binário, pode-se obter seu valor como caracter ASCII.

A principal fraqueza deste método é que se o atacante interceptar a mensagem, ele pode decodificar facilmente se ele conhecer a sequência de cromossomos codificantes utilizados como chave de sessão.

5. Método proposto

Com base no estudo das metodologias já propostas pelos autores acima citados, foi construído um algoritmo de criptografia baseado em DNA, de forma simétrica. O algoritmo possui 4 passos para encriptar a mensagem, a saber, gerar a matriz de substituição, gerar a chave, transformar a mensagem em binário e enfim com a matriz, a chave e o texto em claro, gerar o criptograma. Estes passos geram tanto a chave quanto o criptograma em formato de sequência de DNA, foram feitos utilizando a biblioteca BioJava.

Para simular a criação e manipulação dessas sequências, foram utilizadas algumas operações do DNA-Pascal, no fim da encriptação e no início da decriptação. Que falam da concatenação das sequências, a síntese das mesmas e a recuperação.

- Passo 1: Transformar a mensagem em binário

Para formatar os dados da mensagem de forma a respeitar a limitação de vocabulário imposta pelo alfabeto utilizado no DNA, assim como nos algoritmos propostos na seção anterior, foi feito um método para fazer esta transformação, através do método getBytes () do Java. Isso retorna, para cada caracter da mensagem um grupo de 8 bits, correspondente ao código ASCII. Como ilustrado na tabela abaixo.

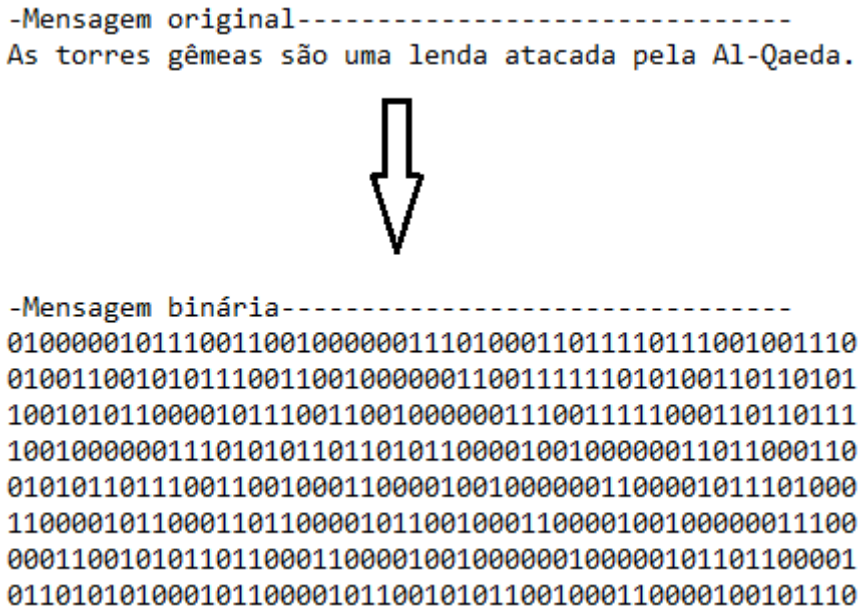


Figura 1. Transformação da mensagem original para binário

Este conteúdo em binário será utilizado pelo próximo passo. Este conteúdo binário é considerado uma String binária, que cresce muito de acordo com o tamanho da mensagem a ser codificada, como pode ser percebido. Uma das vantagens de se usar o DNA para encapsular mensagens é que a compactação de informações neste formato é excelente, possibilitando guardar uma grande quantidade de informações num volume muito pequeno.

- Passo 2: Gerar matriz de substituição

O BioJava entende as sequências como um SymbolList, ou seja, uma lista de símbolos ou uma sequência de bases, os símbolos referem-se a cada base da sequência. Logo, a matriz de substituição nada mais é do que um array de sequências de comprimento 4, que variam seus conteúdos a cada rodada de encriptação. Uma vez gerada, ela é usada para encriptar e decriptar uma mesma mensagem. Logo uma mesma chave, para uma mesma mensagem pode não retornar o mesmo resultado, por causa da aleatoriedade dos conteúdos das posições da matriz. Abaixo, encontra-se um exemplo de matriz gerada.

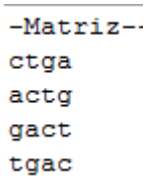


Figura 2. Exemplo de matriz de substituição gerada pelo algoritmo

Antes de alimentar a matriz, é feita uma lista de SymbolList, com os valores de sequências possíveis em cada linha. Na hora de preencher esta matriz, os índices da lista auxiliar são embaralhados, e pega-se o que está na primeira posição a cada rodada, e elimina-se a sequência do índice escolhido na lista auxiliar. Assim na primeira vez possui 4 possibilidades, na segunda existem 3 e assim por diante. Pois para fazer o polialfabético funcionar para um mesmo índice de coluna tem que haver elementos distintos nas linhas.

- Passo 3: Gerar chave

Para gerar a chave, para ter mais confiabilidade na aleatoriedade do conteúdo da chave, usando o mecanismo apresentado no segundo algoritmo simétrico proposto em Terec et Al, foi utilizada a funcionalidade de leitura de arquivos de sequências em FASTA, para obter o conteúdo das sequências. Cada arquivo contém sequências contendo acima de 15.000 bases.

Para saber qual arquivo usar e qual o valor de índice a ser pego, usando o SecureRandom, do pacote Java.Security, foram gerados números inteiros aleatórios seguros, dentro do limite de arquivos e dentro do limite do tamanho da sequência existente em cada um, para escolher qual sequência ler e qual o índice do começo da chave. O comprimento da chave escolhido foi de 256.


```
-Chave-------------------------------------------
attgattcatactctatatgttattattttgtatgcgatgacaacagaatatattatcatgctc
cttttgtgaatctcattcataatataaagtataaatttgtgattttgctttaatttgaaatatt
aatttcaaatatgttatcacaatttgatacaaactattgacagtaaatctgtggattaagtaat
gtcttagtaggtattgggaaaatttgaaactagtaacatggaggaatattgtcattgtttattt
```

Figura 3. Chave gerada para o exemplo

A chave acima foi gerada por um arquivo que continha 17.220 bases. O índice não dá para determinar, pode ser qualquer um que esteja entre o valor do comprimento da sequência lida

- Passo 4: Gerar criptograma

Tendo a matriz de substituição, a mensagem em binário e a chave vindos dos métodos anteriores, utiliza-se a matriz para substituir cada par de 0s e 1s, vindo do resultado do passo de transformação da mensagem para binário, pelo índice correspondente da linha da matriz de substituição em que o caracter da chave corresponder ao elemento da 3ª coluna.

Por exemplo, supondo que, para o primeiro par, o valor da mensagem binária é 01, então será substituído por algum símbolo da 2ª coluna da matriz, para saber qual linha que será usada para substituição, pega-se o caracter da chave para esta posição, "a" por exemplo, e compara seu valor com os símbolos da 3ª coluna, no caso a linha escolhida será a 4ª (levando em consideração e matriz mostrada no passo 2), e então este par 01 será substituído por "g". A ordem é 00 para a 1ª coluna, 01 para a 2ª coluna, 10 para a 3ª coluna e 11 para a 4ª coluna. Esta é a lógica utilizada na encriptação.

```
-Criptograma-------------------------------------
ctggacaggttagtgattccgatagcaaccgactaaaaagacgcaaatgtacg
cggcttctaagagaggctgaatcgaagactagtgccgcggatcatattagtga
cgcccagccaccaaagttgcatccggcgaggatgtttggagtaacctaaatta
tcacctcgagtacgtcttattgaatttcctgcgaaacccaacagaaaacttat
```

Figura 4. Criptograma gerado de acordo com as informações anteriores

Com o criptograma e a chave prontos, usa-se as operações do DNA-Pascal, para simular a criação e a concatenação dessas informações numa cadeia de DNA de fato. De acordo com as operações da tabela de operações principal, vista na seção que falava sobre o DNA-Pascal, abaixo está representado um algoritmo básico para terminar a encriptação, com as respectivas explicações das operações.

begin

t1 := IN(m1) => inicializando para o criptograma

t2 := IN(m2) => inicializando para a chave

t := t1.a => Right adding, adicionando um stopper escolhido e combinado com o receptor, para saber onde começa a chave e onde termina o criptograma. Por exemplo, uma sequência fixa de bases.

T_final := t U t2 => Obtida a mensagem final a ser sintetizada por um sequenciador e entregue ao receptor.

end

Para voltar à mensagem original, as operações em DNA-Pascal reversas terão de ser feitas primeiro, a fim de ler separadamente a sequência da chave e a do criptograma. A seguir entra-se outro algoritmo, o de volta. Também com a explicação das respectivas operações utilizadas.

begin

t := IN(m1) => Inicializando com a sequência obtida pelo receptor

t1 := \t => Subtraindo tudo o que estiver do lado esquerdo do stopper escolhido para separar criptograma de chave, obtendo então o criptograma

t_chave := SX(t, x) => Extraindo o stopper (x) do que restou do corte da sequência, obtendo assim a chave.

end

Tendo a matriz de substituição sido previamente revelada ao emissor, e de acordo com as operações acima, lido as sequências do criptograma e da chave, é possível decriptar e mensagem, utilizando agora o BioJava. Com as operações inversas dos passos apresentados anteriormente.

Agora, ao invés de trocar os pares de 0s e 1s para os símbolos do alfabeto do DNA, os símbolos vão ser trocados pelos pares, seguindo a mesma ideia de se o caracter da chave for o mesmo da terceira coluna de alguma linha, esta que foi encontrada servirá de referência em relação aos símbolos que estão armazenados nela, para trocar para o par de 0 e 1 de acordo com o símbolo que está no criptograma.

Com a string binária aplica-se um outro método, para voltar aos caracteres originais da mensagem, ou seja, ler de 8 em 8 bits transformando de volta em char e concatená-los, para obter a mensagem inteira novamente.

6. Aplicação do método em análise forense

Uma das principais etapas de análise forense é a preservação das provas encontradas, ou seja, fazer com que os dados contidos no material que foi levado a exame de peritos sejam inalterados. Para isso é usada a cadeia de custódia que é um documento que registra a história daquela prova cronologicamente. E ela deve assegurar a proteção e idoneidade da prova, a fim de evitar contradições em relação a sua procedência e estado inicial, pois qualquer suspeita pode anulá-la e impactar no processo de investigação [ALMEIDA 2011].

Para garantir a preservação de evidências, alguns métodos discutidos em Almeida envolve a replicação dos dados em uma cópia para que a análise seja feita em cima da cópia; gravação em mídias ópticas; utilização de outro computador caso os dados sejam muito grandes etc.

Algumas das principais preocupações nesta etapa é garantir o sigilo e a integridade dos dados. Para ter certeza de que os dados não tenham sido alterados ou substituídos até concluir o inquérito. Neste ponto, para verificar a integridade utiliza-se alguns mecanismos como o cálculo realizado a partir de funções de autenticação unidirecionais como o hash, que a partir de uma entrada de qualquer tamanho, gera uma

sequência pequena de bits. E ela é muito utilizada para verificação de integridades pois qualquer alteração, por menor que esta seja, gera um hash completamente diferente. Sendo recomendado hashs acima de 128 bits, para não ter colisão e a partir de dois dados diferentes obter o mesmo valor de hash.

Para se ter sigilo pode-se utilizar o método proposto para criptografar tanto os dados quanto o hash, a fim de que sejam preservadas de forma segura e que possam ser validadas novamente pelo hash. Então, como proposto em Schneier, pode-se utilizar também método de criptografia simétrico para criptografar os hashs. E como o método baseado em fitas de DNA, possui uma grande aleatoriedade, os documentos poderão ser mantidos em segurança e intactos.

7. Conclusões

O método proposto necessita da utilização de máquinas específicas, geralmente encontradas em laboratórios modernos, mas foram utilizados procedimentos simples para tais máquinas, significando uma maior velocidade de leitura e descoberta de informação pelas pessoas autorizadas e que mesmo assim dão um nível de segurança muito alto, pois não há uma substituição direta do binário para as bases.

Além disso, a informação é compactada numa escala muito alta, praticamente imperceptível, o que também pode ser considerado como um tipo de esteganografia. Afinal, como o propósito da esteganografia é esconder a existência de uma mensagem, através da síntese de informações criptografadas em fitas que está a nível microscópico, este propósito é alcançado. Além disso, o método proposto agrega uma criptografia antes de obter a sequência.

Outro aspecto deste método é que dentre várias possíveis aplicações ele pode ser de grande valia para a análise forense na preservação do sigilo e da integridade dos dados contidos nas provas, através de seu método de criptografia e esteganografia.

Os avanços nesta área estão sendo concretizados aos poucos, mas sabe-se que pelo grande poder de armazenamento do DNA, as chaves e os criptogramas podem crescer numa escala inimaginável, o que ajuda a dificultar cada vez mais a complexidade de suas gerações.

Referências

ALMEIDA R. N. (2011) “Perícia Forense Computacional: Estudo das técnicas utilizadas para coleta e análise de vestígios digitais”. Trabalho de conclusão de curso, Faculdade de Tecnologia de São Paulo.

BHOIR Y. R.; MATHANGI R. DNA CRYPTOGRAPHY with BINARY STRANDS. https://www.academia.edu/920735/DNA_Cryptography_using_Binary_Strands, Julho.

ISAIA, E. (2004) “Uma Metodologia para Computação com DNA”. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul.

LIPTON, R. J. (1994) “Speeding Up Computations via Molecular Biology”, In: Princeton University, New Jersey. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.7037, Julho.

Schneier B. (1996) “Applied Cryptography, Second Edition: Protocols, algorithms, and Source Code in C (cloth)”, In: John Wiley & Sons, Inc., ISBN 0471128457.

TEREC R. et Al. (2011) “DNA Security using Symmetric and Asymmetric Cryptography”, In: International Journal on New Computer Architectures and Their Applications, 1(1), 34-51, ISSN 2220-9085.

Clonagem de Cartões Bancários

Gustavo G. Parma, Amilton Soares Júnior

Setor Técnico-Científico – Superintendência Regional de Polícia Federal em MG
Rua Nascimento Gurgel, 30, CEP 30.430-340 – Belo Horizonte – MG – Brazil
{parma.ggp,soares.asj}@dpf.gov.br

Abstract. *This article describes aspects related to the forensic analysis of non-authorized skimming electronic devices installed in ATMs, aimed to clone the magnetic stripe data of bank cards. Initially, information about bank card related financial frauds in Brazil are presented, followed by details concerning the magnetic stripe data format. The article then focuses in presenting the main non-authorized devices usually found in practice and the associated forensic analysis to decode the extracted data from these devices.*

Resumo. *Este artigo apresenta aspectos relacionados aos exames forenses realizados em dispositivos eletrônicos clandestinos instalados de forma camuflada em terminais de autoatendimento bancário, para a clonagem de cartões com tarja magnética. Inicialmente são apresentadas informações sobre os valores envolvidos em fraudes financeiras com cartões bancários no país e as características das tarjas magnéticas desses cartões. Em seguida são mostrados os principais tipos de dispositivos clandestinos utilizados pelos fraudadores e a metodologia de análise forense para a decodificação dos dados extraídos.*

1. Introdução

De acordo com Lobato (2014), no estado de Minas Gerais as fraudes nas compras em lojas virtuais utilizando cartões bancários clonados ou roubados atingiu a cifra de R$ 20 milhões em 2013. No Brasil, o total no mesmo período foi de R$ 350 milhões. Tais montantes teriam sido calculados pela empresa *ClearSale*. No Brasil, ainda de acordo com a reportagem, considerando-se as cinco regiões do país, tem-se uma média de 3,62% de fraudes no total de negócios fechados em lojas virtuais. Em 2013, o comércio eletrônico no país movimentou R$ 28,8 bilhões.

De acordo com a empresa *Serasa Experian* (2014), os setores de comércio e serviço registraram 152.907 tentativas de fraude em fevereiro de 2014, ou seja, uma tentativa de fraude a cada 15,8 segundos, o que representa uma alta de 3,2% em relação a fevereiro de 2013. Mais especificamente, esse total de tentativas distribuiu-se com 37,3% no setor de telefonia, 31,7% no setor de serviços e 20,6% no setor bancário. Já o segmento de varejo registrou 8,2% das tentativas de fraude.

As principais fraudes contra o sistema financeiro nacional envolvem o uso não autorizado de cartões de crédito na aquisição de bens ou serviços, a falsidade ideológica para obtenção de financiamentos, a falsidade ideológica para a abertura de contas bancárias e a falsidade ideológica na abertura de empresas. Todos estes casos passam pelo roubo de identidade, no qual dados pessoais não autorizados são utilizados por criminosos para firmar negócios sob falsidade ideológica ou para obter crédito com a intenção de não honrar os pagamentos. Por dados pessoais entende-se quaisquer

registros de identificação federais ou estaduais (número de CPF, RG, título eleitoral), endereço residencial e/ou comercial, dados bancários e financeiros (banco, agência, conta, cartão de crédito/débito), dentre outros. Como resultado do roubo de identidade, tem-se dívidas e problemas processuais para a pessoa cujos dados foram roubados e prejuízos para lojas, empresas e, de forma geral, para o sistema financeiro nacional.

São diversas as formas de se obter os dados pessoais de um indivíduo. Desde o furto de documentos (já de posse do usuário ou ainda no sistema de entrega) até a utilização de dispositivos eletrônicos camuflados, podendo ainda ser por roubo, *phishing*, uso de *malwares*/*trojans*, buscas na Internet e uso de engenharia social, dentre outros. Este artigo pretende abordar uma forma de roubo de identidade em particular, que é a clonagem de cartões bancários utilizando dispositivos camuflados em terminais de autoatendimento bancário.

2. Tarja Magnética em Cartões Bancários

A tarja magnética presente no reverso dos cartões bancários é composta por três trilhas, nas quais são gravados dados segundo parâmetros físicos estabelecidos pela norma ISO/IEC 7811. Em termos de tipo e codificação, os dados presentes na trilha 1 e trilha 2 obedecem a ISO/IEC 7813, enquanto que os dados da trilha 3 obedecem a ISO/IEC 4909.

A trilha 1 pode conter até 79 caracteres alfanuméricos, codificados em 7 bits, incluindo paridade. Constitui-se em uma trilha apenas para leitura. Os caracteres são divididos em um *Start Sentinel* (SS) "%", um *Format Code* (FC) "B" (para instituições bancárias e financeiras), o *Primary Account Number* (PAN) contendo até 19 dígitos, dos quais os 6 primeiros dígitos são o *Issue Identification Number* (IIN), um *Field Separator* (FS) "^", o nome do proprietário contendo de 2 a 26 caracteres, outro FS "^", a data de vencimento do cartão com quatro dígitos (YYMM), o *Service Code* (SC) com 3 dígitos, o *Discretionary Data* (DD), um *End Sentinel* (ES) "?" e um dígito de *Longitude Redundancy Check* (LRC). A trilha pode conter, ainda, o *Country Code* (CC) com 3 dígitos e o *Pin Verification Value* (PVV) com 5 dígitos, dependendo da bandeira do cartão.

A trilha 2, por sua vez, é a única efetivamente necessária e essencial para a realização de transações financeiras eletrônicas. Ela contém até 40 caracteres BCD, codificados em 5 bits, incluindo paridade, e também constitui-se em uma trilha apenas para leitura. Os dados desta trilha são divididos em um SS ";", o PAN contendo até 19 dígitos, dos quais os 6 primeiros são o IIN, o FS "=", a data de vencimento do cartão com quatro dígitos (YYMM), o SC com 3 dígitos, o DD, o ES "?" e um dígito de LRC. A trilha pode conter, ainda, o CC e o PVV, dependendo da bandeira do cartão.

A trilha 3 é raramente utilizada, por se tratar de uma trilha de leitura e escrita. Elá é composta por até 107 caracteres BCD, codificados em 5 bits, incluindo paridade. Ela possui um SS ";", o FC com 2 dígitos, o PAN em até 19 dígitos, um dígito de FS "=", diversos dígitos de controle, o ES "?" e um dígito de LRC.

3. Clonagem de Cartões Bancários

Uma operação financeira realizada com utilização de cartão bancário envolve, geralmente, dois elementos distintos para segurança. O primeiro elemento é "algo que o usuário possui", sedo no caso o próprio cartão bancário ou, de forma mais exata, os dados nele contidos. O segundo elemento é "algo que o usuário sabe", que é a senha

associada àquele cartão bancário. Algumas transações financeiras menos seguras requerem apenas os dados do cartão bancário, sem a necessidade da senha do usuário, como é o caso da maioria das compras pela Internet.

Conforme mencionado, a trilha 2 é aquela realmente necessária para a realização de transações financeiras eletrônicas utilizando cartões bancários. Desta forma, a clonagem de cartões bancários segue uma lógica que envolve duas etapas distintas. Na primeira etapa os dados da trilha 2 do cartão alvo da fraude são obtidos. Na segunda, os dados obtidos são copiados para um segundo suporte plástico, o qual é então utilizado nas transações fraudulentas pelos criminosos. Ressalta-se que, na primeira etapa, dependendo da estratégia utilizada para a obtenção dos dados, a senha associada pode também ser ou não obtida.

Os dados da trilha 2 de um cartão bancário podem ser obtidos por diversas maneiras, como o furto ou desvio do cartão original (antes da chegada do mesmo até seu proprietário), a utilização de dispositivos eletrônicos camuflados, *phishing* e uso de *malwares*/*trojans*, dentre outros artifícios. Este artigo tem como foco o caso do uso de dispositivos eletrônicos camuflados.

Neste tipo de artifício, conforme indicado pelo próprio nome, dispositivos eletrônicos clandestinos (ou um conjunto deles), artesanais ou não, são instalados de forma camuflada em equipamentos autorizados normalmente utilizados pelos usuários para a realização de transações financeiras eletrônicas. Como exemplos de equipamentos autorizados podem ser citados os terminais de autoatendimento bancário (ATM – *Automatic Teller Machine*), os terminais *Point of Sale* (POS) e os terminais PIN PAD. Os dispositivos clandestinos são instalados de forma que o usuário comum não perceba a presença dos mesmos e, em geral, não interferem no funcionamento normal dos equipamentos. Este tipo de dispositivo camuflado é popularmente chamado de “chupa-cabra”. Exemplos dos mesmos, com diversos graus de sofisticação e acoplados a diferentes tipos de equipamentos autorizados, são mostrados nas Figuras 1 a 5.

Na Figura 1 é possível visualizar a parte frontal completa de um simulacro a ser instalado em um terminal de autoatendimento. Neste caso, o acesso aos botões de comando do ATM por parte do usuário é prejudicado e, na estratégia da fraude, o simulacro simula o funcionamento do ATM. Contudo, na realidade, a operação financeira não se concretiza verdadeiramente. A Figura 2 mostra um simulacro semelhante ao do caso anterior, porém, com partes separadas para se encaixarem sobre o monitor, sobre o teclado e sobre o leitor de cartões, estratégia esta escolhida dependendo das possibilidades oferecidas pelo *layout* do ATM verdadeiro. Tal como na situação anterior, o funcionamento do ATM é prejudicado.

A Figura 3 mostra uma mini-CPU e seus acessórios, os quais são instalados dentro do ATM. A Figura 4 apresenta a visão externa e interna de um POS adulterado. Já a Figura 5 apresenta um simulacro que é sobreposto no leitor de cartões magnéticos existente no ATM. Em todos estes casos, o usuário não é capaz de perceber, pelo funcionamento do equipamento, a presença dos dispositivos eletrônicos camuflados. Além disso, o objetivo específico dos dispositivos camuflados nesses casos é obter os dados da trilha 2 da tarja magnética de cartões bancários. Ressalta-se, ainda, que à exceção do dispositivo mostrado na Figura 5, a lógica das conexões eletrônicas permite que seja obtida diretamente, também, a senha do usuário.

Os dispositivos camuflados mostrados nas Figuras 1, 2 e 5 são personalizados para um modelo (*layout*) de ATM em particular. Alterando-se o modelo do ATM é necessário alterar o formato do simulacro. Porém, suas partes eletrônicas constituintes permanecem as mesmas. Em todos os modelos apresentados como exemplos, os dispositivos eletrônicos camuflados possuem sua própria cabeça magnética para a leitura dos dados da tarja dos cartões bancários. Entretanto, em outros modelos os fraudadores utilizam a própria cabeça magnética já existente no equipamento autorizado, antes de sofrer adulteração.

Figura 1. Frente falsa a ser sobreposta em um ATM.

Figura 2. Partes falsas a serem sobrepostas em um ATM.

Figura 3. Mini CPU e acessórios para serem instalados dentro de um ATM.

Figura 4. Visão externa e interna de um POS adulterado.

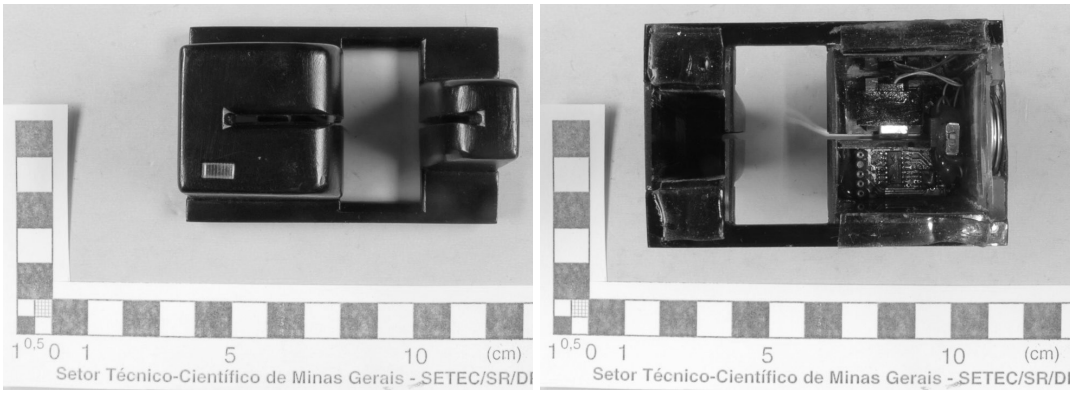


Figura 5. Anverso e reverso de um simulacro a ser sobreposto em um ATM.

4. Análise Forense dos Dispositivos Eletrônicos Camuflados

Conforme apresentado, os dispositivos camuflados utilizam uma cabeça magnética (própria ou não) para a leitura dos dados da trilha 2 da tarja dos cartões bancários. Tal cabeça magnética é conectada a um circuito eletrônico de apoio que, por sua vez, possui componentes eletrônicos para o condicionamento do sinal elétrico proveniente da leitura e abriga um microcontrolador e uma memória, geralmente serial. A Figura 6 apresenta um tipo de circuito eletrônico o qual é utilizado em dispositivos camuflados como aquele mostrado na Figura 5.

O circuito eletrônico pode possuir, ainda, alguma indicação luminosa, posicionada de tal forma a simular o funcionamento normal do equipamento (leitor de tarjas magnéticas do ATM) quando da inserção do cartão bancário. O microcontrolador pode armazenar os dados lidos da trilha 2 diretamente na memória serial existente no circuito, enviar os dados para armazenamento em outra mídia, tal como o disco rígido de um notebook, ou, alternativamente, enviar os dados por uma interface aére (utilizando uma conexão *bluetooth*, por exemplo). Para a extração dos dados capturados e armazenados na memória do circuito, os fraudadores geralmente utilizam a interface serial existente no microcontrolador.

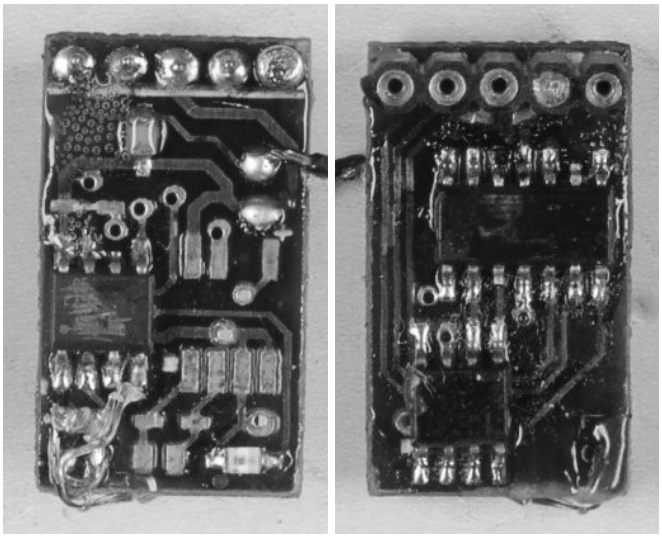


Figura 6. Anverso e reverso de circuito típico de um chupa-cabra.

A análise forense tem como objetivo central produzir prova material com a busca de informações sobre a materialidade do ato, sua autoria e sua dinâmica. No caso em questão, procura-se determinar o estado de funcionamento do dispositivo, se o mesmo é capaz de obter os dados da trilha 2 de cartões magnético e, finalmente, se existem dados da trilha 2 armazenados na memória de seu circuito. Essa análise forense robustece a evidência encontrada no local de crime.

Para tentar dificultar a análise do sistema, os fraudadores raspam a identificação do modelo tanto do microcontrolador quanto da memória serial existente no sistema, informações essas normalmente impressas sobre o invólucro do *chip*. Além disto, são configuradas senhas para acesso ao microcontrolador, via interface serial. Outro artifício geralmente utilizado pelos fraudadores para mascarar os dados obtidos da trilha 2 de cartões bancários, dificultando a análise forense, é a codificação dos caracteres antes do

armazenamento dos dados na memória.

Para o acesso aos dados que porventura tenham sido capturados pelo dispositivo, tentativas de quebra de senha por força-bruta mostraram-se ineficazes na maioria dos casos examinados. A solução encontrada, então, para a análise dos dados presentes na memória serial de circuitos, tal como o da Figura 6, é a extração da memória da placa de circuito impresso e leitura dos dados diretamente por um leitor universal (processo denominado "*dump* da memória"). Para tal, considerando que os fraudadores raspam a indicação de marca e modelo de memória, é feita uma análise do circuito buscando-se determinar o modelo genérico da memória. São realizadas, então, tentativas de leitura da memória até que se obtenha os dados armazenados.

Posteriormente, os dados obtidos e salvos em arquivo são analisados e devidamente decodificados, de forma a serem obtidos dados compatíveis com o formato de armazenamento na trilha 2 de cartões bancários. Para auxiliar nesta decodificação, desenvolveu-se um *software* no Setor Técnico-Científico da Superintendência de Polícia Federal em Minas Gerais. Tal *software* permite análises estatísticas nos dados da memória, buscando-se determinar o código utilizado antes do armazenamento dos dados na memória. O *software* permite, adicionalmente, a aplicação de códigos anteriormente já identificados ou, alternativamente, a aplicação de um novo código na decodificação dos dados. Até o momento já foram identificados diversas codificações distintas, geralmente de substituição simples de alfabeto, utilizadas em dispositivos eletrônicos camuflados. Contudo, ressalta-se que já foram encontrados caos de codificação multi-alfabeto.

5. Conclusão

Foram apresentados neste artigo aspectos diversos relacionados a fraudes executadas no âmbito do sistema financeiro nacional por meio do roubo de informações de cartões bancários e utilização destas de maneira desautorizada e ilegal, o que causa prejuízos da ordem de dezenas de bilhões de reais ao país por ano. Além dos aspectos técnicos sobre o formato dos dados contidos nas tarjas magnéticas dos cartões, focou-se na análise dos principais tipos de dispositivos eletrônicos camuflados utilizados pelos fraudadores para roubo de informações no caso em questão e na metodologia de análise forense para a decodificação dos dados extraídos dos dispositivos encontrados.

O Setor Técnico-Científico da Superintendência de Polícia Federal em Minas Gerais desenvolveu-se um *software* que permite análises estatísticas dos dados extraídos da memória dos dispositivos camuflados, auxiliando o Perito Criminal na determinação do código utilizado no embaralhamento dos dados originais da trilha 2 de cartões bancários. Já foram identificados diversas codificações, tanto de substituição simples de alfabeto quanto de multi-alfabeto.

A utilização da metodologia e do *Software* desenvolvido possibilitam, atualmente, uma taxa aproximada de 100% na decodificação dos dados presentes na memória de dispositivos eletrônicos camuflados tais como o mostrado na Figura 6.

Referências

Lobato, P. H. (2014) “Tentativas de Compras com Cartões Clonados em Lojas Virtuais Somaram R$ 350 mi no País em 2013”, http://www.em.com.br/app/noticia/economia/2014/04/20/internas_economia,520826/tentativas-de-compra-com-cartoes-clonados-em-lojas-virtuais-somaram-r-350-mi-no-pais-em-2013.shtml, acessado em agosto 2014.

Serasa Experian (2014) “Tentativas de Fraude contra o Consumidor sobem 3,2% em Fevereiro, de acordo com Indicador da Serasa Experian”, http://noticias.serasaexperian.com.br/tentativas-de-fraude-contra-o-consumidor-sobem-32-em-fevereiro-de-acordo-com-indicador-da-serasa-experian/, acessado em agosto 2014.

International Organization for Standardization and International Electrotechnical Commission (ISO/IEC). Padrão 7813:2006 – Identification Cards – Financial Transaction Cards, 2006.

International Organization for Standardization and International Electrotechnical Commission (ISO/IEC). Padrão 7811 – Identification Cards – Recording Technique.

International Organization for Standardization and International Electrotechnical Commission (ISO/IEC). Padrão 4909:2006 – Identification Cards – Financial Transaction Cards – Magnetic stripe data content for track 3, 2006.

Uso de Funções de *Hash* em Forense Computacional

Marcos A. C. Corrêa Júnior, Ruy J. Guerra B. de Queiroz

[1]Universidade Federal de Pernambuco (UFPE)

{maccj, ruy}@cin.ufpe.br

Abstract. *The increased use of computing devices and the access to services through these devices, has contributed to the growth of cybercrime victims. In this paper, is showed the importance of combating computer crime through computer forensics. The focus of this work is on technical aspects, but are placed legal points that provide general instructions for how to proceed in computer forensics for admissibility of digital evidence in court. The article also makes a comparison between different hash functions, these functions have properties that make them useful in computer forensic related to the integrity of the digital evidence collected during an investigation.*

Resumo. *A ampliação do uso de dispositivos computacionais juntamente com o acesso a serviços por meio desses dispositivos, tem contribuído para o crescimento de vítimas de crimes cibernéticos. Nesse artigo, mostra-se a importância do combate ao crime por meio da forense computacional. O foco do trabalho é em aspectos técnicos, mas são colocados pontos jurídicos que fornecem instruções gerais de como proceder em forense computacional para que uma prova eletrônica seja aceita em um tribunal. O artigo também realiza uma comparação entre diferentes funções de hash, essas funções apresentam propriedades que as tornam úteis à forense computacional no que diz respeito a integridade das provas eletrônicas coletadas no decorrer de uma investigação.*

1. Introdução

Com a utilização crescente de computadores pessoais, *smartphones*, *tablets* e vários outros dispositivos computacionais para realizar transações bancárias, trocas de informações valiosas, comércio, os criminosos expandiram sua área de atuação. No mundo virtual há possibilidade (ou crença) de permanecer anônimo, grande e rápido retorno financeiro, efemeridade das provas, possibilidade de cometer o crime em qualquer parte do mundo, ausência ou escassez de leis que punam os infratores, falta de capacidade do Estado em prender e punir os criminosos.

O Instituto Ponemon [Institute 2013] que realizou um estudo independente para a *HP Enterprise Security* mostrou o crescimento dos crimes cibernéticos e concluiu que o impacto financeiro causados por ataques cibernéticos aumentou, em 2013, 30% em relação a 2012. Ainda através desse estudo, constatou-se que houve em média 1,4 ataques bem sucedidos por semana a cada empresa analisada.

A forense computacional consiste na aplicação de métodos científicos em dispositivos computacionais durante a realização de investigações com o intuito de encontrar informações relevantes para casos de interesse judicial. [Anderson et al. 2012] afirma que

deveria se gastar menos dinheiro na antecipação do crime (antivírus, *firewall*, IDS, etc) e mais dinheiro nas investigações, a fim de encontrar e punir os criminosos.

Para que o esforço de encontrar os criminosos não seja desperdiçado, as evidências encontradas devem ser preservadas, mantidas sem alteração, íntegras. A importância de se manter a integridade delas ocorre devido ao fato de que essas evidências serão usadas pelo juiz para formar sua convicção e prolatar a sentença. Se não houver mecanismos que assegurem que os dados coletados permanecem inalterados, esses dados podem ter sido fraudados sem que ninguém saiba, logo não se poderia presumir que se tratam de dados autênticos. A ausência dessa garantia de integridade permite que o réu argua falsidade dos dados, ainda que esses não tenham sido alterados.

Há na criptografia opções de mecanismos que permitem demonstrar que os dados permanecem íntegros. Duas linhas destacam-se: funções de *hash* e códigos de autenticação da mensagem baseados em cifra de bloco. O trabalho propõe-se a mostrar funções de *hash* que podem ser empregadas em forense computacional para a verificação da integridade de dados.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a forense computacional e seus princípios. A Seção 3 apresenta os trabalhos relacionados e como o trabalho proposto se diferencia deles. A Seção 4, a aplicação dos *hashes* criptográficos para verificação da integridade. A Seção 5, um estudo comparativo avaliando as funções de *hash*. Finalmente, a Seção 6 apresenta as conclusões deste trabalho.

2. Forense Computacional

A forense computacional é a aplicação de métodos científicos para responder às investigações e às questões de interesse legal. A aplicação desses métodos científicos é realizada em dispositivos computacionais com a finalidade de determinar se ele foi ou está sendo utilizado para fins ilegais, se foi alvo da ação ilegal ou ainda se serve de local de armazenamento de material relacionado a algum ato ilícito. A metodologia empregada na área é considerada ainda muito recente, apesar de já fazer algum tempo que há procedimentos definidos para examinar e coletar evidências de dispositivos computacionais (tablets, celulares, notebooks, computadores, etc), de mídias de armazenamento (pen drive, hd, ssd, fitas, discos óticos, etc) e de conexões destes dispositivos. Na figura 1 é apresentado um modelo genérico com os passos seguidos em uma investigação de forense computacional.

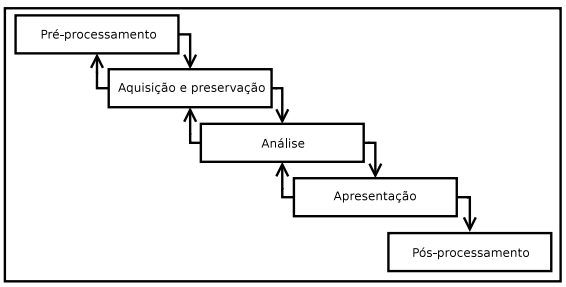


Figura 1. Modelo de Investigação de Forense Computacional [Yusoff et al. 2011]

Os trabalhos de perícia, em geral, precisam de maior apoio e investimentos por parte do Poder Público. O documento Diagnóstico Perícia Criminal [SENASP 2012] emi-

tido pela Secretaria Nacional de Segurança Pública - SENASP, mostra a falta de estruturas minimamente padronizadas dos institutos de perícia criminal existentes no Brasil. Ainda segundo o documento, se falta pessoal, equipamentos e capacitação, falta mais do que tudo uma gestão adequada, sem a qual o país seguirá carente desse serviço tão relevante o qual, em conjunto com outras provas, contribui para que a autoridade judicial forme a sua convicção, seja para absolver ou para condenar, protegendo direitos e reduzindo a impunidade.

A lei 12.735/2012 [Brasil 2012] acrescenta ao Código Penal Brasileiro um artigo que obriga os órgãos de polícia judiciária a se estruturarem para o combate à ação delituosa em rede de computadores, dispositivo de comunicação ou sistema informatizado.

2.1. Prova eletrônica

Na doutrina, Paulo Rangel conceitua prova como sendo: "O meio instrumental de que se valem os sujeitos processuais (autor, juiz e réu) de comprovar os fatos da causa, ou seja, os fatos deduzidos pelas partes como fundamento do exercício dos direitos de ação e de defesa " [Rangel 2007]. A prova é usada para demonstrar a existência de fatos alegados. No direito brasileiro não há uma hierarquia entre as provas [Lima 2004]. No entanto, reconhece-se que algumas provas tem um maior grau de certeza e confiabilidade, como é o caso dos exames de DNA em ações de investigação de paternidade.

A tecnologia tem sido protagonista nessa transição em que se busca maior grau de certeza e confiabilidade da prova. Será considerada "prova eletrônica" aquelas cujo local de armazenamento seja eletrônico, cujo elemento armazenado consista de sequência de números binários que reconhecidos pelo computador, representam uma informação [Lessa 2010].

As provas eletrônicas não gozam de tão boa reputação quanto à prova técnica de exame de DNA. Para Demócrito Reinaldo Filho, alguns problemas da prova eletrônica são: "a informação em formato eletrônico é dinâmica, o mero ato de ligar ou desligar um computador pode alterar a informação que ele armazena"; "os computadores quando em funcionamento reescrevem e deletam informação, quase sempre sem o conhecimento específico do operador" e "a informação armazenada eletronicamente, ao contrário de textos escritos em papel, pode se tornar incompreensível quando separada do sistema que a criou" [Reinaldo Filho 2006]. Renato Blum afirma que é uma questão de extrema relevância a validade dos documentos eletrônicos, visto que por meio de recursos técnicos é possível alterar documentos digitais sem deixar vestígios [Blum 2012]. Breno Lessa em seu artigo intitulado "A inviabilidade das provas digitais no processo judiciário" [Lessa 2010] afirma: "a integridade do documento eletrônico só poderia ser confirmada se fosse possível assegurar que o documento não foi atacado, não foi alterado ou adulterado, mas isso é praticamente impossível, principalmente nos computadores pessoais".

Para a utilização da prova eletrônica, há vários aspectos que são questionados no âmbito jurídico. O Supremo Tribunal Federal já entendeu como válida a utilização de arquivo eletrônico [Silva 2011], além disso a Medida Provisória 2.200/2001, que instituiu a Infra-Estrutura de Chaves Públicas Brasileira - ICP-Brasil, trata o arquivo eletrônico como documento e garantiu-lhe a presunção de veracidade, se ele for assinado digitalmente com certificado digital emitido pela ICP-Brasil ou outro certificado aceito pelas partes [Brasil 2001].

A aceitação da prova eletrônica, como foi visto, é perfeitamente possível, mas uma série de procedimentos devem ser tomados para que a prova não seja descartada por suspeita de sua adulteração, por exemplo. A força probante de um documento digital será maior quanto maior for definida sua autoria, sua autenticidade e sua integridade.

2.2. Técnicas de forense computacional

O processo de investigação forense, a princípio, dividia-se em duas técnicas conhecidas como *live analysis* e *post-mortem*. [Huebner et al. 2007] sugeriu uma terceira técnica que se baseava na ideia de recriar o cenário a ser investigado. Além da definição da estrutura geral dos processos (passos básicos da análise forense computacional) que deve ser seguida, outra questão que ainda suscita discussão em forense computacional é quanto a melhor abordagem a se adotar, ao se deparar no local das buscas com uma máquina ainda em funcionamento, três abordagens são possíveis:

1. realizar alguns procedimentos aproveitando-se da máquina ligada (técnica de *live analysis* e de reprodução do ambiente);
2. desligá-la através da interrupção do fornecimento de energia (*post-mortem* 1);
3. proceder desligamento administrativo normal (*post-mortem* 2).

Em [Auler et al. 2011], afirma-se que em casos de busca e apreensão de computadores é recomendável realizar uma coleta de dados voláteis e de dados lógicos de áreas protegidas por senha enquanto o computador ainda estiver ligado, antes de efetuar o desligamento da máquina para apreensão tradicional.

2.2.1. *Live Analysis*

Esta técnica se presta a coletar também as informações que possuem alta ordem de volatilidade. A *live analysis* altera os dados armazenados na memória, aceita-se que estas alterações são inevitáveis, mas há alterações que não são admitidas. Uma manipulação mal feita pode gerar dados falsos ou omitir informações, ainda há o risco de que uma conexão aberta esteja sendo usada para apagar evidências contidas no dispositivo enquanto ele está sendo analisado por peritos.

2.2.2. *Post-mortem*

Abordagem tradicional, tem como premissa a preservação das evidências armazenadas em mídias não voláteis. Os equipamentos estão previamente desligados ou, se estiverem ligados, serão desligados após encontrados. A fonte de informação é o conteúdo de informação contido nas mídias não voláteis, é mais fácil garantir que as mídias não sofram alteração.

2.2.3. Reprodução do Ambiente

Uma nova abordagem em investigação forense computacional é tentar recriar o sistema computacional e seu ambiente. Mesmo que a abordagem não seja admitida em um tribunal, ela é muito útil, poderá determinar as razões da brecha de segurança causada por um *bug*, quando não há intenções criminosas, por exemplo [Huebner et al. 2007].

3. Trabalhos Relacionados

Em [Karyda and Mitrou 2007], são discutidos aspectos técnicos e jurídicos a respeito da análise forense e da utilização dos dados coletados em um eventual processo. Dentre os aspectos técnicos tidos como problemas para a forense são citados a diversidade e heterogeneidade da infraestrutura, além de barreiras que impedem os investigadores de acessar as fontes de dados. Quanto aos aspectos jurídicos, afirma-se que todos os atributos necessários para uma prova comum devem também ser necessários para provas eletrônicas. A prova tem que ser admissível (não pode ser contrária a nenhum princípio legal nem procedimento processual), deve ser irrefutável e autêntica, além de estar ligada diretamente ao evento criminoso. Afirma-se ainda que para ser bem sucedido na demonstração da integridade e autenticidade do material em um tribunal, métodos ou técnicas padronizadas devem ser seguidas na execução dos passos da forense computacional. Em [Huebner et al. 2007] são mencionados testes ou ferramentas usadas pelos tribunais para determinar o mérito científico da evidência apresentada. Cita que a suprema corte americana sugeriu critérios para determinar se a ciência é confiável e, portanto, admissível, um dos critérios era, por exemplo, se a evidência se baseava em uma técnica ou teoria testada.

[Ke et al. 2011] discute o uso de algoritmos de *hash* em forense computacional para assegurar que a prova eletrônica é autêntica. Menciona-se no artigo que por conta da natureza das funções de *hash* qualquer mudança, por menor que seja, na *string* de entrada dará como saída um resultado totalmente diferente. Nesse trabalho, mostra-se como as funções de *hash* podem ser empregadas em forense computacional para aperfeiçoar a confiabilidade da evidência digital. [Kumar et al. 2012] afirma que ferramentas forenses costumam utilizar as funções de *hash* SHA-1 e MD5 para verificar se um conjunto de dados foi alterado, o artigo discute a importância da utilização do valor de *hash* em forense computacional no tocante à integridade da prova eletrônica.

O trabalho [Oluwasegun et al. 2014] afirma que acusações de crimes os quais utilizam provas eletrônicas estão se tornando mais eficazes devido a melhorias nos métodos de investigação e de coleta de evidências. Menciona-se nesse trabalho que um processo investigativo bem feito e organizado em etapas padronizadas permite que as evidências coletadas consigam associar o suspeito ao crime e permite que essas evidências sejam admitidas em um tribunal.

Esses trabalhos não levam em consideração doutrina, legislação e jurisprudência nacionais, além do mais nenhum deles pesquisou as funções de *hash* MD5, SHA-1, família SHA-2 [FIPS 2012] e família SHA-3 [FIPS 2014].

4. Integridade e hashes criptográficos

Para que a prova eletrônica seja considerada de fato confiável e afaste a suspeita de que possa ter acontecido qualquer espécie de adulteração, é necessário que a integridade da informação seja assegurada. Integridade faz parte da tríade CIA (do inglês *confidentiality, integrity, availability*) que costumeiramente é associada à segurança da informação. A integridade de dados é uma propriedade ou serviço de segurança que garante que os dados recebidos ou armazenados não foram alterados ou destruídos de forma não autorizada. De maneira mais completa, pode-se dizer ainda que a integridade de dados é a garantia de que os dados recebidos estão exatamente da mesma forma em que foram enviados por uma entidade autorizada e não contêm modificações, inserção, exclusão ou repetição [Stal-

lings 2013]. Os mecanismos voltados para a integridade de dados são usados para garantir que dados de um sistema computacional ou de redes não sejam comprometidos. Há várias ameaças à integridade de dados que podem ocasionar o comprometimento das informações em um sistema computacional ou em redes de comunicações, esses comprometimentos podem ser divididos em dois grandes grupos: o primeiro grupo inclui as falhas benignas que compreendem formas ocasionais de modificação dos dados (por exemplo uma interferência eletromagnética natural forte, ruídos no canal de comunicação que tenham capacidade de inverter um bit em um arquivo armazenado ou em uma transmissão, respectivamente); o segundo, inclui as falhas maliciosas que ocorrem pela ação intencional (por exemplo por meio da inserção de um vírus de computador que altera o conteúdo dos dados).

4.1. Medidas para obtenção de integridade

Desde os mecanismos de integridade empregados em canais de comunicação por Hamming [Hamming 1950] até os dias atuais, com relação não só aos canais de comunicação mas também com relação às grandes bases de dados, integridade é um aspecto importante de segurança da informação. A integridade permite mostrar que aquele dado é confiável, não foi alterado. Há vários mecanismos projetados para apoiar a integridade dos dados, dentre eles incluem-se:

- Cópias de segurança - é o arquivamento feito de modo que as informações possam ser recuperadas caso tenham sido alteradas, seja de forma intencional ou não;
- Códigos detectores de erros - uma pequena alteração no arquivo de entrada (como a mudança de um só bit) acarrete um valor de saída diferente;
- Códigos corretores de erros - projetados para que pequenas alterações possam ser detectadas e automaticamente corrigidas.

Em forense computacional precisa-se com frequência utilizar as cópias de segurança e mecanismos que nos permitam verificar alterações - como fazem os códigos detectores de erros. Todas esses mecanismos mencionados dependem de certo grau de redundância. No caso das cópias de segurança (backup), a aplicabilidade em forense computacional se dá para que toda a análise pericial se realize sobre a cópia e não sobre a mídia original, para isso é necessário que a cópia do conteúdo seja feita para um dispositivo de igual ou maior capacidade de armazenamento e que, notadamente, seja de mesma natureza (exemplos: de HD para HD, de *blu-ray* para *blu-ray*, etc). A RFC 3227 [Brezinski and Killalea 2002] recomenda ainda que se o objetivo final for realizar uma análise forense, a cópia de segurança deverá ser realizada bit-a-bit e todo o trabalho pericial de análise será desenvolvido sobre a cópia porque a análise quase sempre irá alterar os tempos de acesso aos arquivos. Entre os mecanismos que permitem alertar sobre a ocorrência de pequenas alterações, que apenas verificam se a informação original foi alterada, há os códigos de autenticação de mensagens (MACs - *Message Authentication Codes*) baseados em cifras de bloco e também as funções de *hash*. Os códigos de autenticação de mensagens baseados em cifras de bloco fogem do escopo de nosso trabalho, as funções de *hash* serão vistas na subseção 4.2.

4.2. Funções de *hash* ou funções de resumo

As funções de *hash* são muitas vezes conhecidas também como funções de resumo. Ferguson [Ferguson et al. 2012] chama atenção para uma ambiguidade: o termo “funções

de *hash*" é também empregado para funções de mapeamento usadas para acessar tabelas *hash*, uma estrutura de dados usada em muitos algoritmos. Essas funções homônimas apresentam propriedades similares às funções de *hash* criptográficas, mas há enorme diferença entre as duas. As funções de *hash* possuem propriedades de segurança criptográficas específicas. A função *hash* de mapeamento para tabelas *hash* possuem requisitos muito mais fracos.

A definição mais comum para funções de *hash* é:
"uma função que mapeia uma *string* de bits de comprimento arbitrário para uma *string* de bits de comprimento fixo." [Dang 2012]

Dependendo da propriedade ou das propriedades que a aplicação criptográfica necessite, há três propriedades desejáveis:

1. Resistência à colisão
2. Resistência à primeira inversão
3. Resistência à segunda inversão

As funções de *hash* têm uma grande variedade de usos em segurança, a seguir são mencionados alguns desses usos:

- *Autenticação de mensagens* - mecanismo ou serviço usado para verificar a integridade de uma mensagem, permite assegurar que a informação recebida é igual à mensagem enviada (sem modificação, inserção, supressão ou *replay*);
- *Assinaturas digitais* - o valor de resumo de uma mensagem é cifrado com a chave privada do emissor, qualquer usuário que possua a chave pública pode verificar a integridade da mensagem que é associada com a assinatura digital;
- *Arquivo de senhas* - um resumo de uma senha fica armazenado em um arquivo do sistema operacional em vez de se armazenar a senha propriamente dita, caso o arquivo de senha seja violado, o atacante só conseguirá obter o *hash* da senha;
- *Detecção de intrusos e vírus* - para cada arquivo F do sistema, armazena-se também o *hash* $H(F)$, se houver qualquer em F, será percebida;
- *Construção de funções pseudoaleatórias (PRF) ou construção de gerador de números pseudoaleatórios (PRNG)* - para geração de chaves simétricas.

Na aplicação em Forense Computacional, o uso de funções de *hash* é fundamental porque há muita desconfiança dos operadores do direito com relação à confiabilidade que se pode atribuir a documentos e provas eletrônicas [Lessa 2010]. O emprego de funções de *hash* em forense ocorre principalmente na autenticação de mensagens e nas assinaturas digitais.

Funções de *hash* são unidirecionais (do inglês, *one-way*), é fácil calcular o valor de *hash* a partir de uma entrada qualquer de comprimento arbitrário, em contrapartida é difícil retornar ao valor anterior da mensagem a partir de um dado valor de *hash*. A situação ideal é que todo valor de *hash* gerado fosse realmente único para cada valor de entrada, isso normalmente não é possível pois em nossas funções de *hash*, o contradomínio (os vários valores de *hash* possíveis de serem gerados) geralmente é muito menor do que o seu domínio (possíveis valores de entrada).

5. Avaliação das funções de hash

5.1. Ataques de força bruta e ataques criptoanalíticos

Pode-se agrupar os ataques relacionados com as funções hash em duas categorias:

- Ataques por força bruta - a capacidade de uma função de *hash* resistir a ataques por força bruta depende do tamanho do código *hash* produzido na saída, hoje tanto os valores apresentados pelo MD5 (saída de 128 bits) quanto os valores apresentados pelo SHA-1 (saída de 160 bits) são considerados fracos.
- Ataques por criptoanálise - buscam explorar alguma propriedade do algoritmo para realizar algum ataque diferente de uma busca exaustiva. Um ataque criptoanalítico para ser considerado bem sucedido deve apresentar um esforço menor que o esforço exigido por um ataque de força bruta. A partir disso, pode-se enunciar que uma função de hash é resistente a ataques criptoanalíticos se, diante do melhor ataque criptoanalítico a força empregada para realizar tal ataque criptoanalítico é maior ou igual a força utilizada em ataques de força bruta.

5.2. Ataque de extensão de comprimento

Um ataque preocupante é o ataque criptoanalítico devido à vulnerabilidade de extensão do comprimento. As funções de *hash* MD5, SHA-1 e SHA-2 apresentam a vulnerabilidade que permite o ataque de extensão do comprimento e que leva a problemas reais, problemas esses que poderiam e deveriam ser evitados [Ferguson et al. 2012].

Para entender melhor o ataque de extensão, pense em uma mensagem m que é separada em blocos: $m = m_1, \ldots, m_k$ e resumido para um valor ***H***. Escolha uma mensagem ***m'*** que é separada em blocos: $m' = m_1, \ldots, m_k, m_{k+1}$. Como os primeiros *k* blocos de m' são idênticos aos *k* blocos da mensagem m, o valor de hash *h(*m*)* é meramente o valor de hash intermediário depois de *k* blocos no cálculo de *h(*m'*)*. Tem-se que *h(*m'*)* = *h'(h(*m*),*m_{k+1}*)*. Quando se usar MD5 ou qualquer função da família SHA, tem-se que escolher m' cuidadosamente para incluir o campo de preenchimento e comprimento, mas isto não é um problema pois o método de construção desses campos é conhecido.

O problema da extensão de comprimento existe porque não há processamento especial no fim do cálculo da função *hash*. O resultado é que *h(*m*)* provê informação direta sobre os estados intermediários depois dos primeiros *k* blocos de m'. Isto é certamente uma surpresa para uma função que se acreditava que era de mapeamento aleatório. De fato, esta propriedade imediatamente desqualifica - de acordo com a definição de segurança de funções *hash* - as funções MD5, SHA-1 e SHA-2 mencionadas acima e que serão vistas nas seções 5.3, 5.4 e 5.5, respectivamente [Ferguson et al. 2012]. Tudo que um distinguidor tem que fazer é construir pares (m, m') e checar esta relação. Certamente em uma função de *hash* ideal não se encontraria esta relação. O ataque requer apenas alguns cálculos *hash*, então é muito rápido este ataque. Como esta vulnerabilidade pode ser explorada? Imagine um sistema onde Alice envia uma mensagem a Bob e quer autenticar enviando h($X||m$), onde X é um segredo conhecido só por Bob e Alice, e m é a mensagem. Se h fosse uma função de *hash* ideal, isto faria um bom sistema de autenticação, mas com a vulnerabilidade da extensão do comprimento, Eve pode adicionar texto a mensagem m, e atualiza o código de autenticação para coincidir com a nova mensagem. Um sistema de autenticação que permite a um indivíduo externo modificar a mensagem é, claramente, inútil para nós.

Na chamada de trabalhos para a competição de escolha do SHA-3, um dos requisitos estabelecidos pelo NIST (acrônimo em inglês de *National Institute of Standards and Technology*) é que SHA-3 não fosse susceptível ao ataque da extensão do comprimento.

O ataque de extensão do comprimento permite portanto inclusão de informação extra ao final da mensagem sem que esta alteração seja perceptível, o que compromete totalmente a integridade da mensagem e abre uma possibilidade de que a informação seja comprometida sem que se saiba. Tornar uma informação digital facilmente alterável sem deixar vestígio, a torna também inválida para uso em um tribunal, por conta da suspeita que se pode levantar de adulteração desta prova [Lessa 2010].

5.3. MD5

O MD5 (Message Digest 5) é uma função de hash criada por Ronald Rivest e especificado na RFC 1321 [Rivest 1992]. Desenvolveu-se a partir da função de *hash* MD4 e recebeu reforços adicionais contra ataques. O MD4 era bem rápido, deu origem a outras importantes funções de *hash*, mas foi quebrado. O MD5 se tornou uma das funções de *hash* mais famosas ainda é possível encontrar menção dele em vários livros atuais e encontrá-lo em execução em sistemas atuais, apesar de ser uma função ultrapassada que também foi quebrada [Wang and Yu 2005]. Desde a criação do MD5 já foram encontradas diversas vulnerabilidades e ninguém deveria estar utilizando o MD5 na atualidade [Stevens et al. 2009] [Xie and Feng 2009]. O MD5 deve ser considerado um algoritmo ultrapassado e inútil [Schneier 2008].O primeiro passo para calcular o *hash* com MD5 é pegar uma mensagem de tamanho arbitrário e dividi-la em blocos de 512 bits. Ao último bloco devem ser adicionados bits de preenchimento e também o comprimento da mensagem. A mensagem de entrada é resumida em uma mensagem de 128 bits.

Usando o paradoxo do aniversário, podia-se facilmente encontrar colisões com 2^{64} tentativas, o que quer dizer que se teria que calcular aproximadamente 2^{64} vezes para conseguir o mesmo resumo. Porém os avanços dos ataque criptoanalíticos tornaram a situação do MD5 ainda pior com ataque de colisão com um custo computacional equivalente a $2^{49.8}$ chamadas a função de compressão MD5 [Stevens 2012], além de ataques com custo computacional teórico de cerca de 2^{16} chamadas a função de compressão [Stevens et al. 2009] e, Xie e Feng anunciaram que sob certas circunstâncias poderia-se chegar a um ataque muito rápido, com custo computacional teórico de cerca de 2^{10} chamadas a função de compressão [Xie and Feng 2009]. No entanto, eles não publicaram um ataque real ainda.

5.4. SHA-1

O SHA (*Secure Hash Algorithm*) foi publicado em 1993 pela NSA (acrônimo em inglês de *National Security Agency*) e NIST como um padrão de processamento de informações federais (do inglês, *federal information processing standard* - FIPS de número 180). Foram descobertas fraquezas na versão inicial publicada em 1993 que ficou conhecida como SHA-0 (não foram divulgados detalhes sobre as fraquezas na época), foi então que uma versão revisada foi publicada em 1995 hoje conhecida como SHA-1 (FIPS 180-1). O SHA-1 é uma versão mais moderna do SHA e que ganhou maior notoriedade. O SHA-1 produz saídas maiores que as do MD5, suas saída são de 160 bits e estima-se que se teria que calcular 2^{80} vezes para se conseguir a mesma saída a partir de entradas distintas [Stallings 2013].

SHA também é uma função de *hash* baseada na função MD4, por conta dessa herança comum há algumas características comuns entre SHA-1 e MD5, porém SHA-1 acaba sendo mais lenta por conta de seu projeto mais cuidadoso com a segurança. O

SHA-1 assim como o MD5 foi alvo de uma série de pesquisas que buscam definitivamente quebrá-lo. Apesar do projeto mais cuidadoso, infelizmente, SHA-1 também é inseguro e as fraquezas criptográficas encontradas fizeram com que esta função não fosse mais recomendada para a maior parte dos usos criptográficos após 2010 [Wang et al. 2005] [Ke et al. 2011]. Apesar dos ataques ao SHA-1, isto não quer dizer que seja possível recuperar a mensagem original em um curto espaço de tempo [Ke et al. 2011]. Segundo cálculos apresentados por Schneier [Schneier 2012], mas que foram realizados por Jesse Walker, só em 2018 o poder computacional necessário para encontrar um ataque de colisão alcançará valores razoáveis US$173k, que poderiam ser pagos por uma organização criminosa.

Com estes trabalhos, prova-se que foram encontradas formas mais rápidas de encontrar saídas de mensagens de resumo SHA-1 iguais, por meio de entradas diferentes, isto pode ser conseguido em um tempo factível, logo o SHA-1 não é mais computacionalmente seguro [Ke et al. 2011]. Para mitigar os problemas encontrados nas funções MD5 e SHA-1 foram propostas em [Ke et al. 2011] as seguintes contramedidas:

- a inclusão de informações adicionais por meio de *strings*, que acrescentariam rótulos de tempo como parte da operação, dessa forma não só se reduz a possibilidade de colisões, mas também aumenta-se a confiabilidade da evidência;
- a utilização de novos mecanismos de *hash* como SHA-2 e SHA-3.

Atualmente o ataque conhecido mais eficiente consegue encontrar colisões usando bem menos esforço do que 2^{80} chamadas a função SHA-1, o ataque de Stevens publicado em 2013 consegue encontrar colisões em $2^{57.5}$ [Stevens 2013]. Por fim, [Dang 2012] afirma que, após o término de 2013, aplicações de assinatura digital não deverão usar o SHA-1.

5.5. SHA-2

Em 2001, o NIST publicou um rascunho de padrão (*draft*)contendo três novas funções de *hash SHA* (SHA-256, SHA-384 e SHA-512), em agosto de 2002 foi publicado o padrão FIPS 180-2. Em 2004 o padrão FIPS 180-2 foi modificado, em 2008 nova modificação, o documento passa a ser a FIPS 180-3, nele ainda consta a antiga função SHA-1 e as novas funções de *hash* SHA que ficaram conhecidas como SHA-2. A RFC 6234 [Eastlake and Hansen 2011] também especifica SHA-1 e SHA-2, e mostra implementações em C.

A versão mais atual da família SHA-2 foi publicada em março de 2012 na FIPS 180-4 [FIPS 2012], neste padrão, além da família SHA-2 (com as funções SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 e SHA-512/256) também está presente a antiga função SHA-1.

As funções de *hash* da família SHA-2 possuem saídas de 224, 256, 384, 512, 224 e 256 bits de acordo com seus nomes SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 e SHA-512/256 respectivamente. A estrutura destas funções é muito similar a do SHA-1. A família SHA-2 é a substituta natural do SHA-1, apesar de não ter passado por um longo escrutínio público, as funções SHA-2 foram examinadas pela NSA, que tem *expertise*. A segurança dos algoritmos de *hash* é um pouco mais detalhada no documento [Dang 2012].

O cuidado que o SHA-2 tem com a segurança tem um preço, o *hash* com a função SHA-256 é muito mais lento do que com o SHA-1. Para mensagens longas calcular um *hash* com o SHA-256 pode levar tanto tempo quanto cifrar a mensagem com AES ou Twofish, ou até um pouco mais de tempo [Ferguson et al. 2012].

Com tantos diferentes valores de saída que forneceriam segurança de até 2^{256}, levando-se em conta o paradoxo do aniversário, com a possibilidade de usar 512 bits de saída, a família SHA-2 parece oferecer uma segurança inatacável, no entanto o SHA-2 usa a mesma estrutura e operações matemáticas dos seus antecessores, e isso é um motivo de preocupação, pensando nisto, o NIST anunciou uma nova competição para produzir a nova geração SHA que viria a ser chamada de SHA-3.

O SHA-512 é a função recomendada para ser usada em procedimentos periciais no Brasil [SENASP 2013], apesar de parecer a melhor opção por ter resistência a colisão de 2^{256}, resistência à primeira inversão de 2^{512} e resistência a segunda inversão entre 2^{394} e 2^{512} [Dang 2012]. A função é vulnerável a ataques genéricos da família SHA, dentre eles o *ataque de extensão de comprimento*. Por ocasião da abertura da competição para a escolha do SHA-3 o NIST, ciente da vulnerabilidade de extensão do comprimento que atinge as versões SHA-1 e SHA-2, colocou como requisito de segurança para as concorrentes, possuir resistência contra ataque de extensão do comprimento [Gligoroski 2011] [Kayser 2007].

Por herdar muitos dos ataques genéricos já descobertos para o SHA-1, devido à similar forma de construção e operações matemáticas, o SHA-2 deve ser descontinuado e substituído pelo SHA-3 [AlAhmad and Alshaikhli 2013].

5.6. SHA-3

Em 2007 o NIST publicou [Kayser 2007] documento em que anunciou a competição para escolher a terceira geração de funções *hash*, neste documento estavam previstos requisitos mínimos para disputar a competição. A função de *hash* vencedora, escolhida em outubro de 2012, foi escolhida entre as finalistas BLAKE, Grostl, JH, Keccak e Skein. Nesta disputa saiu vencedora a Keccak, uma função de *hash* criptográfica projetada por Guido Bertoni, Joan Daemen, Michael Peeters e Gilles Van Assche [Al-shaikhli et al. 2013].

Keccak, a função SHA-3 escolhida, suporta os mesmos valores de resumo (saídas) que a família SHA-2 (224, 256, 384 e 512), mas sua estrutura interna difere significativamente de toda a família SHA e permite níveis de alta segurança. Quanto a estrutura e ao projeto a função de compressão tem uma estrutura composta de sete permutações Keccak-f para construir a chamada função esponja. Com seu projeto, Keccak pode gerar diferentes comprimentos de saída e ao mesmo tempo ter flexibilidade para aumentar o nível de segurança. A função SHA-3 tem alto nível de paralelismo com uma fraca difusão dos bits para as diferentes porções, por conta disto, a função tem também um comportamento diferente em sua operação criptoanalítica, a função esponja opera de forma que há uma fase de absorção (entrada) e há uma fase de compressão (saída).

Em [Al-shaikhli et al. 2013], afirma-se que a função de *hash* Keccak não herda vulnerabilidades genéricas da família SHA, devido a construção do tipo esponja.Ainda quanto a segurança, Keccak não tem a vulnerabilidade de extensão do comprimento diferentemente do SHA-1 e do SHA-2 [Bertoni et al. 2014]. Até o término deste artigo o padrão definitivo da família SHA-3 (FIPS 202) ainda não havia sido publicado, porém já está disponível desde maio de 2014 uma versão preliminar (*draft*) do padrão FIPS 202 [FIPS 2014].

A Tabela 1 resume a proteção oferecida pelas diferentes funções de *hash* que ganharam notoriedade e foram analisadas neste artigo. A existência de ataques criptoa-

Tabela 1. Comparativo de parâmetros de funções de *hash*

	MD5	SHA-1	SHA-2						SHA-3			
Tamanho da Saída	128	160	224	256	384	512	512/224	512/256	224	256	384	512
Tam. Máx. da Mensagem	2^{64}	2^{64}	2^{64}	2^{64}	2^{128}	2^{128}	2^{128}	2^{128}	∞	∞	∞	∞
Tamanho do Bloco	512	512	512	512	1024	1024	1024	1024	1152	1088	832	576
Tamanho da Palavra	32	32	32	32	64	64	64	64	64	64	64	64
Número de Rounds	64	80	64	64	80	80	80	80	24	24	24	24
Ataques	*X/0*	*X/0*	*0*	*X/0*	*0*	*X/0*	*0*	*0*				

nalíticos (de resistência à colisão, de resistência à primeira inversão ou de resistência à segunda inversão) que exigem esforço menor do que um ataque de força bruta, é simbolizado com *X* (no caso do SHA-256 e SHA-512 são versões com passos reduzidos), a possibilidade de ataques genéricos à construção Merkle-Damgard (como o ataque de extensão do comprimento, que demonstrou sucesso em ataque ao Flickr [Duong and Rizzo 2009]) é simbolizado com *0*.

6. Conclusões

Este artigo apresentou argumentos para o emprego de funções de *hash* em procedimentos forenses computacionais. O objetivo principal da proposta foi demonstrar que é possível provar com muita acurácia que a informação coletada foi conservada tal qual a originalmente encontrada, fazendo uso de mecanismos conhecidos como funções de *hash* - amplamente utilizadas para verificação da integridade de dados. É importante ressaltar, no entanto, que não basta utilizar qualquer função de *hash*, pois se for mal utilizada poderá passar uma falsa sensação de integridade da evidência coletada, é preciso usar corretamente as funções de acordo com os estudos científicos mais recentes e confiáveis. Adicionalmente, a proposta comparou diferentes funções de *hash* e mencionou embasamentos legais para que a integridade dos dados coletados sejam uma preocupação durante o desenvolvimento de uma investigação forense computacional. O trabalho desenvolvido recomenda a utilização da função de *hash* SHA-3 que apresenta quatro comprimentos de saída distintos que podem ser escolhidos de acordo com o nível de segurança que se deseje. A escolha da função SHA-3 é mais indicada do que a da SHA-512 que é recomendada pelo documento entitulado Procedimento Operacional Padrão publicado pela SENASP/MJ em 2013 [SENASP 2013] a fim de orientar o profissional de perícia da área de informática. Conforme foi explicado na subseção 5.6, a função SHA-512 é susceptível a ataques de extensão de comprimento (fraqueza bem conhecida da construção Merkle-Damgard) que permite a um indivíduo malicioso anexar texto a mensagem original, e atualizar a *tag* de validação para combinar com a nova mensagem, se um indivíduo maliciosos é capaz de fazer isso, esta função vulnerável a ataques de extensão não é útil. O SHA-3 por atender aos requisitos de segurança contidos em [Kayser 2007], não é vulnerável a ataques de extensão e seu uso deve ser recomendado no documento de Procedimento Operacional Padrão publicado pela SENASP/MJ. A recomendação de usar o SHA-3 se justifica para que as provas eletrônicas que usem este mecanismo como forma de garantir a integridade não só sejam aceitas em um tribunal, mas também utilizem o que há de mais moderno em termos de função de *hash*.

Referências

Al-shaikhli, I. F., Alahmad, M. A., and Munthir, K. (2013). Hash Function of Finalist SHA-3: Analysis Study. *Information Technology (IJACSIT)*, 2(2).

AlAhmad, M. A. and Alshaikhli, I. F. (2013). Broad view of cryptographic hash functions.

Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M., Levi, M., Moore, T., and Savage, S. (2012). Measuring the Cost of Cybercrime. In *WEIS*.

Auler, P., de Melo, L. P., de Deus, F. E. G., and de Sousa Jr, R. T. (2011). Uma Nova Abordagem em Apreensão de Computadores.

Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G. (2014). The Keccak Sponge Function Family. `http://keccak.noekeon.org/`.

Blum, R. M. O. (2012). Internet e os Tribunais. *Revista da Academia de Peritos*, 1(1).

Brasil (2001). MEDIDA PROVISÓRIA. Nº2.200, DE 24 DE AGOSTO DE 2001.

Brasil (2012). LEI Nº12.735, DE 30 DE NOVEMBRO DE 2012.

Brezinski, D. and Killalea, T. (2002). RFC 3227 - Guidelines for evidence collection and archiving. `https://www.ietf.org/rfc/rfc3227.txt`.

Dang, Q. H. (2012). SP 800-107. Recommendation for Applications Using Approved Hash Algorithms.

Duong, T. and Rizzo, J. (2009). Flickr's API Signature Forgery Vulnerability.

Eastlake, D. and Hansen, T. (2011). RFC 6234 - US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF). `http://www.ietf.org/rfc/rfc6234.txt`.

Ferguson, N., Schneier, B., and Kohno, T. (2012). *Cryptography Engineering: Design Principles and Practical Applications*. John Wiley & Sons.

FIPS, N. (2012). 180-4: Secure hash standard (SHS).

FIPS, N. D. (2014). SHA-3 standard: Permutation-based hash and extendable-output functions. *DRAFT FIPS*, 202.

Gligoroski, D. (2011). Length Extension Attack on Narrow-Pipe SHA-3 candidates. In *ICT Innovations 2010*, pages 5–10. Springer.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. *Bell System technical journal*, 29(2):147–160.

Huebner, E., Bem, D., and Bem, O. (2007). Computer Forensics–Past, Present and Future. *Information security Technical report*, 8(2):32–46.

Institute, P. (2013). 2013 Cost of Cyber Crime Study: Global Report.

Karyda, M. and Mitrou, L. (2007). Internet forensics: Legal and technical issues. In *IEEE 2nd International Workshop on Digital Forensics and Incident Analysis (WDFIA2007)*.

Kayser, R. F. (2007). Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family. *Federal Register*, 72(212):62.

Ke, H.-J., Liu, J., Wang, S.-J., and Goyal, D. (2011). Hash-Algorithms Output for Digital Evidence in Computer Forensics. In *IEEE International Broadband and Wireless Computing, Communication and Applications (BWCCA2011)*, pages 399–404.

Kumar, K., Sofat, S., Jain, S., and Aggarwal, N. (2012). Significance of Hash Value Generation in Digital Forensic: A Case Study. *International Journal of Engineering Research and Development. Available at: http://www. ijerd. com/paper/vol2-issue5 I.*

Lessa, B. M. (2010). A inviabilidade das provas digitais no processo judiciário. `http://jus.com.br/artigos/14555`.

Lima, R. K. d. (2004). Direitos civis e Direitos Humanos: uma tradição judiciária pré-republicana? *São Paulo em Perspectiva*, 18:49 – 59.

Oluwasegun, S., David, O. E., Esther, E., and Victor, O. (2014). Computer forensics for law enforcement. *Journal of Emerging Trends in Engineering and Applied Sciences*.

Rangel, P. (2007). *Direito Processual Penal*. Lumen Juris, 12th edition.

Reinaldo Filho, D. (2006). A exibição da prova eletrônica em juízo–Necessidade de alteração das regras do processo civil.

Rivest, R. (1992). IETF RFC 1321: The MD5 Message-Digest Algorithm.

Schneier, B. (2008). Forging SSL Certificates. `https://www.schneier.com/blog/archives/2008/12/forging_ssl_cer.html`.

Schneier, B. (2012). When Will We See Collisions for SHA-1? `https://www.schneier.com/blog/archives/2012/10/when_will_we_se.html`.

SENASP (2012). *Diagnóstico da Perícia Criminal no Brasil*. SENASP - Depto. de Pesquisa, Análise da Informação e Desenv. de Pessoal em Seg. Pública.

SENASP (2013). *Procedimento Operacional Padrão - Perícia Criminal*. SENASP - Depto. de Pesquisa, Análise da Informação e Desenv. de Pessoal em Seg. Pública.

Silva, L. A. d. (2011). Valor Probante dos Documentos Eletrônicos. *Revista Jurídica do MPRN - Ano 1 v.1 Jun/Dez 2011*, pages 120–143.

Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Pearson Prentice Hall, 6th edition.

Stevens, M. (2012). Single-block collision attack on MD5. *IACR Crypto ePrint Archive*.

Stevens, M. (2013). New collision attacks on SHA-1 based on optimal joint local-collision analysis. In *Advances in Cryptology–EUROCRYPT 2013*, pages 245–261. Springer.

Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D. A., and De Weger, B. (2009). Short chosen-prefix collisions for md5 and the creation of a rogue ca certificate. In *Advances in Cryptology-CRYPTO 2009*, pages 55–69. Springer.

Wang, X., Yin, Y. L., and Yu, H. (2005). Finding collisions in the full SHA-1. In *Advances in Cryptology–CRYPTO 2005*, pages 17–36. Springer.

Wang, X. and Yu, H. (2005). How to break MD5 and other hash functions. In *Advances in Cryptology–EUROCRYPT 2005*, pages 19–35. Springer.

Xie, T. and Feng, D. (2009). How to Find Weak Input Differences for MD5 Collision Attacks.

Yusoff, Y., Ismail, R., and Hassan, Z. (2011). Common phases of computer forensics investigation models. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(3):17–31.

Banco de Dados de Laudos Periciais de Dispositivos Móveis

Alonso Decarli[1], Cícero Lemes Grokoski[1], Emerson Cabrera Paraiso[1], Luiz Rodrigo Grochocki[2], Cinthia O. A. Freitas[1]

[1]Pontifícia Universidade Católica do Paraná (PUCPR) – Escola Politécnica – Programa de Pós-Graduação em Informática (PPGIa)
R: Imaculada Conceição, 1155 – 80.215-901 – Curitiba – PR – Brasil

[2]Instituto de Criminalística do Paraná – Polícia Científica do Paraná
Av. Visconde de Guarapuava, 2652 – 80.010-100 – Curitiba – PR – Brasil

{paraiso, cinthia}@ppgia.pucpr.br, luiz.grochocki@ic.pr.gov.br,
cicero.grokoski@gmail.com, alonsodecarli@gmail.com

Abstract. *Searches related to treatment of extracted digital data from mobile devices require a database created to provide the technical and scientific details of the computer forensics. This paper presents in detail the design of the database based on expert reports from mobile devices to be used as default in the Institutes of Criminology nationwide. The project specifies the input from the reports in XML to the relational structure of the database and also the type of result expected by the public security forces. Furthermore, it demonstrates a practical way, using a case study, as the validation can be performed.*

Resumo. *Pesquisas relacionadas ao tratamento de dados digitais extraídos de dispositivos móveis necessitam de uma base de dados constituída de modo a atender as especificações técnicas e científicas da área de computação forense. Este artigo apresenta detalhadamente o projeto do banco de dados de laudos periciais de dispositivos móveis a ser utilizado como padrão nos Institutos de Criminalística de todo o país. O projeto especifica desde a entrada dos laudos em XML até a estrutura relacional do banco de dados e, ainda, o tipo de resultado esperado por parte das forças de segurança pública. Além disto, demonstra de modo prático, utilizando um estudo de caso, como o tratamento e cruzamento de informações poderão ser realizados.*

1. Introdução

A problemática relacionada aos procedimentos periciais em dispositivos móveis foi apresentada por [Grochocki et al. 2013], tendo como base o Sistema SiCReT, o qual é alimentado por laudos periciais, sendo que tal sistema se vale do uso da ciência da informação para disponibilizar ferramentas que auxiliem os Serviços de Inteligência e Policiamento Preditivo.

A ideia central do sistema SiCReT é que todos os laudos periciais relativos a aparelhos telefônicos alimentem um Banco de Dados com informações consolidadas para apontar e investigar em cada laudo as propriedades e o comportamento da informação de modo a auxiliar no entendimento sobre as forças que governam o fluxo criminoso (Figura 1).

Dentre as principais contribuições que o sistema SiCReT irá proporcionar, tem-se a padronização do formato de arquivo dos laudos periciais, a obtenção de dados estatísticos, a estruturação dos dados extraídos dos dispositivos móveis em um Banco Dados Relacional, organizado de modo a prover estruturas aptas à aplicação de técnicas de mineração de dados. Destaca-se ainda o desenvolvimento de um módulo customizado para instituições que trabalham com Análise Forense de Dispositivos Móveis, subdividido em: extração de características transformando dados brutos em estruturas de dados, *interface web* para visualização dos resultados obtidos e, também, aplicação de técnicas de Mineração de Dados. Todos estes elementos facilitarão aos peritos o tratamento e cruzamento de dados provenientes dos diferentes dispositivos móveis apreendidos nas mais diversas situações. Em longo prazo, espera-se evitar a subjetividade da atividade pericial no tocante ao cruzamento de registros telefônicos.

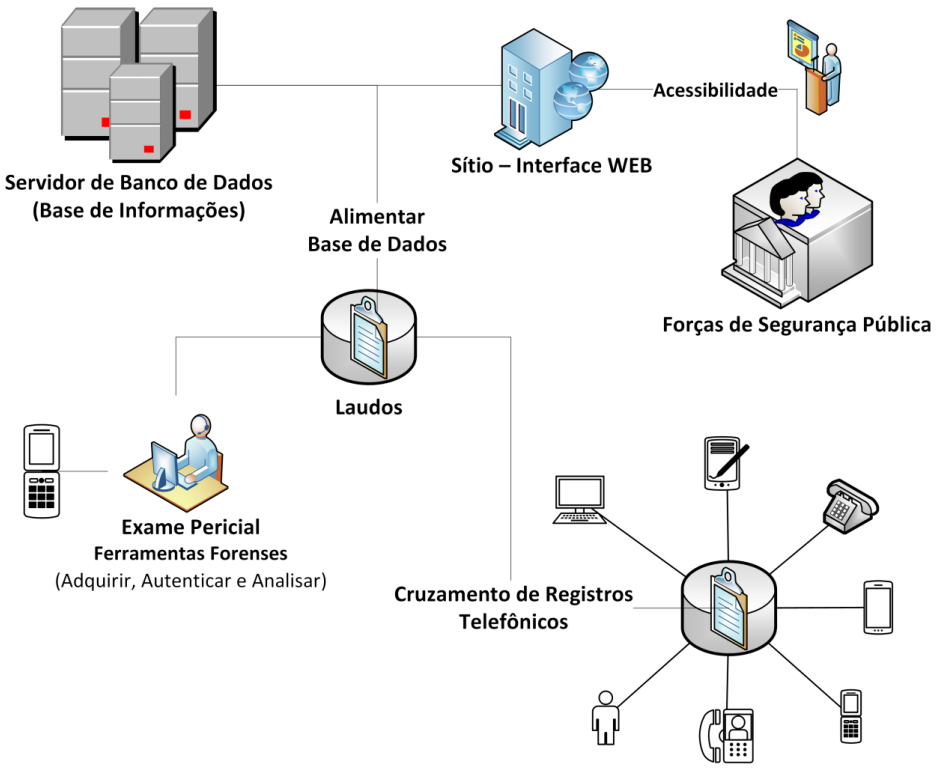


Figura 1 – Visão Geral do Sistema

Este artigo está organizado de tal forma que a Seção 2 resume aspectos técnicos e jurídicos sobre dispositivos móveis e a coleta de dados digitais sob a ótica da computação forense. A Seção 3 apresenta o projeto do Banco de Dados Relacional do Sistema SiCReT detalhando um Estudo de Caso. Na Seção 4 é apresentado um Estudo de Caso visando a realização dos testes de Consistência do Banco de Dados. A Seção 5 apresenta as tecnologias utilizadas até o momento. Por fim, a Seção 6 aborda conclusões e trabalhos futuros.

2. Dispositivos Móveis e Coleta de Dados Digitais

A Computação Móvel proporciona a capacidade de mover fisicamente serviços computacionais junto com os usuários, tornando os dispositivos computacionais sempre presentes, permitindo ao ser humano ter acesso aos recursos oferecidos por um sistema computacional independentemente da sua localização [Weiser 1993].

Na Publicação Especial 800-101 do NIST (*National Institute of Standards and Technology*) [Jansen e Ayres 2007] os autores sugerem que a chave para o sucesso na análise forense de dispositivos móveis é a compreensão das características de *hardware* e *software* dos telefones celulares. Os dados dos assinantes e suas atividades por meio de celulares são muitas vezes uma fonte valiosa de provas em uma investigação. Portanto, para que a produção de provas possa ser realizada, conta-se com um conjunto básico de características, obtido a partir da maioria dos celulares, sendo este conjunto comparável entre diferentes aparelhos. Como exemplos de características podem ser citados: microprocessador, memória ROM (*Read Only Memory*), memória RAM (*Random Access Memory*), módulo de rádio, processador de sinal digital, alto falante, tela, sistema operacional, bateria, PDAs (*Personal Digital Assistants*), GPS (*Global Positioning System*), câmera, entre outros recursos.

A aquisição de dados a partir de um dispositivo pode ser física ou lógica [Jansen e Ayres 2007]. A aquisição física tem vantagens sobre a aquisição lógica, uma vez que permite que os arquivos apagados e alguns dados restantes possam ser examinados, por exemplo, na memória não alocada ou em espaço do sistema de arquivos. Os autores recomendam sempre fazer a aquisição de dados física antes da aquisição lógica. As ferramentas forenses adquirem informações dos dispositivos sem alterar o conteúdo, ou seja, em modo somente de leitura, utilizando equipamentos denominados de *Write Blocker* (ou bloqueadores de escrita). E, ainda, fazendo a geração de *hash* de modo a garantir a integridade dos dados coletados. Tal característica é muito importante perante um Juiz, sendo que cabe ao perito garantir a integridade das provas digitais por ele coletadas ou a ele confiadas.

Neste contexto, a atividade pericial da análise forense de dispositivos móveis inicia-se na aquisição dos dados digitais a partir dos dispositivos móveis (celulares, *tablets*, entre outros) sendo que neste artigo foram estudados os métodos das ferramentas forenses de captura, *hardware/software*, utilizadas pelo Instituto de Criminalística, a saber: Cellebrite UFED e Microsytemation XRY. Ambos realizam a extração de dados em padrão XML (*eXtensible Markup Language*), padrão este que permite descrever diversos tipos de dados, tendo como objetivo facilitar o compartilhamento de informações.

O *Cellebrite UFED* (*Universal Forensic Extraction Device*) *Touch Ultimate* é uma solução composta por *hardware* e *software* proprietário que permite a extração, decodificação, análise e geração de relatórios avançados em termos tecnológicos dos dados de dispositivos móveis, sendo suportados atualmente 7.900 dispositivos diferentes. Conta também com uma interface interativa com tela sensível ao toque e um conjunto de cabos com interface USB e RJ45 que realizam a conexão e comunicação com os dispositivos móveis. Dentre os principais aplicativos que acompanham a solução vale destacar: *Phisical Analyzer* (ferramenta de decodificação, análise e relatórios), *Phone Detective* (*software* que identifica um telefone móvel no início de uma

investigação) e *Reader* (inicialização dos dispositivos em modo somente leitura, permitindo o compartilhamento de informações).

Quando o *Cellebrite UFED* não fornece suporte ao dispositivo a ser analisado, o Instituto de Criminalista conta com o *Microsytemation XRY*, que realiza função semelhante de captura, porém com maior suporte a equipamentos, visto ser uma solução composta por aplicativos (*software)* e equipamentos (*hardware)* que permite aos peritos realizar a extração forense física e lógica de dispositivos móveis. Em sua versão atual, a 6.7, a solução XRY suporta 10.036 dispositivos móveis, sendo que sua principal característica é essa considerável quantidade de dispositivos suportados, devido a esse fator a ferramenta é utilizada em larga escala na área forense.

A Tabela 1 exemplifica informações gerais da captura de dados em dispositivos móveis, com base no *hardware/software Cellebrite UFED e Microsytemation XRY*. A partir de todo o conjunto de dados extraído, os peritos sabem *a priori* que são importantes, para o cruzamento, os dados contidos nos contatos da agenda, nas chamadas (realizadas e recebidas) e mensagens instantâneas trocadas entre celulares distintos. Sabem ainda que alguns modelos de *smartphones* fornecem dados de mensagens eletrônicas (*e-mail*), salas de bate-papo (*chat*), entre outros.

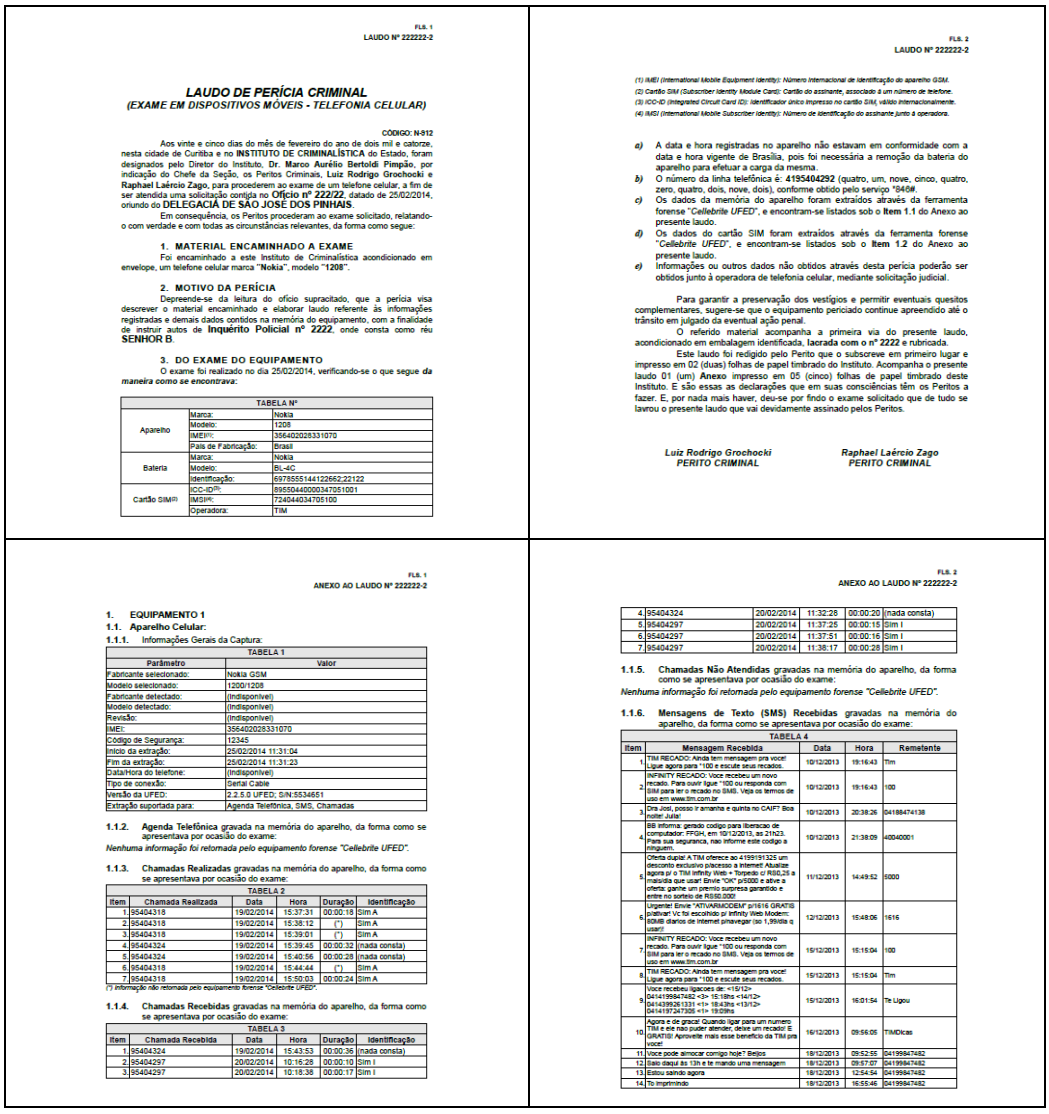
Tabela 1 - Informações Gerais da Captura

Parâmetros
Fabricante selecionado
Modelo selecionado
Fabricante detectado
Modelo detectado
Nome do equipamento
IMEI (*International Mobile Equipment Identity*)
ICCID (*International Circuit Card ID*)
IMSI (*International Mobile Subscriber Identity*)
Endereço Bluetooth
Endereço Wi-Fi
Início da extração
Fim da extração
Data/Hora do telefone
Tipo de conexão
Versão da UFED

Assim, os dados extraídos dos dispositivos móveis passam a compor o que é denominado de Laudo Pericial (Figura 2 – reprodução parcial). Na esfera da metodologia científica em perícia criminal, [Reis 2011] aponta que "um relatório é uma exposição gráfica e geral de um assunto, sendo que compreende desde o planejamento, passando por todos os procedimentos, até chegar à conclusão, materialidade e autoria, incluindo-se os processos metodológicos empregados, recursos, equipamentos e ferramentas".

Diferente de um trabalho técnico comum, o trabalho pericial elege como elemento essencial de sua estrutura a resposta aos quesitos propostos à perícia, a qual pode ser apresentada através de parecer sucinto, apenas com respostas aos quesitos formulados, ou através da exposição detalhada dos elementos investigados, sua análise e

fundamentação das conclusões, além das respostas aos quesitos formulados [Rosa 1999].



FLS. 1
LAUDO Nº 222222-2

LAUDO DE PERÍCIA CRIMINAL
(EXAME EM DISPOSITIVOS MÓVEIS - TELEFONIA CELULAR)

CÓDIGO: N-912

Aos vinte e cinco dias do mês de fevereiro do ano de dois mil e catorze, nesta cidade de Curitiba e no **INSTITUTO DE CRIMINALÍSTICA** do Estado, foram designados pelo Diretor do Instituto, **Dr. Marco Aurélio Bertoldi Pimpão**, por indicação do Chefe da Seção, os Peritos Criminais, **Luiz Rodrigo Grochocki** e **Raphael Laércio Zago**, para procederem ao exame de um telefone celular, a fim de ser atendida uma solicitação contida no **Ofício nº 222/22**, datado de 25/02/2014, oriundo do **DELEGACIA DE SÃO JOSÉ DOS PINHAIS**.

Em consequência, os Peritos procederam ao exame solicitado, relatando-o com verdade e com todas as circunstâncias relevantes, da forma como segue:

1. MATERIAL ENCAMINHADO A EXAME

Foi encaminhado a este Instituto de Criminalística acondicionado em envelope, um telefone celular marca "**Nokia**", modelo "**1208**".

2. MOTIVO DA PERÍCIA

Depreende-se da leitura do ofício supracitado, que a perícia visa descrever o material encaminhado e elaborar laudo referente às informações registradas e demais dados contidos na memória do equipamento, com a finalidade de instruir autos de **Inquérito Policial nº 2222**, onde consta como réu **SENHOR B**.

3. DO EXAME DO EQUIPAMENTO

O exame foi realizado no dia 25/02/2014, verificando-se o que segue ***da maneira como se encontrava:***

TABELA Nº		
Aparelho	Marca:	Nokia
	Modelo:	1208
	IMEI(1):	356402028331070
	País de Fabricação:	Brasil
Bateria	Marca:	Nokia
	Modelo:	BL-4C
	Identificação:	6978555144122662;22122
Cartão SIM(2)	ICC-ID(3):	89550440000347051001
	IMSI(4):	724044034705100
	Operadora:	TIM

FLS. 2
LAUDO Nº 222222-2

(1) IMEI (International Mobile Equipment Identity): Número internacional de identificação do aparelho GSM.
(2) Cartão SIM (Subscriber Identity Module Card): Cartão do assinante, associado à um número de telefone.
(3) ICC-ID (Integrated Circuit Card ID): Identificador único impresso no cartão SIM, válido internacionalmente.
(4) IMSI (International Mobile Subscriber Identity): Número de identificação do assinante junto à operadora.

a) A data e hora registradas no aparelho não estavam em conformidade com a data e hora vigente de Brasília, pois foi necessária a remoção da bateria do aparelho para efetuar a carga da mesma.
b) O número da linha telefônica é: **4195404292** (quatro, um, nove, cinco, quatro, zero, quatro, dois, nove, dois), conforme obtido pelo serviço "*8458#*.
c) Os dados da memória do aparelho foram extraídos através da ferramenta forense "*Cellebrite UFED*", e encontram-se listados sob o **Item 1.1** do Anexo ao presente laudo.
d) Os dados do cartão SIM foram extraídos através da ferramenta forense "*Cellebrite UFED*", e encontram-se listados sob o **Item 1.2** do Anexo ao presente laudo.
e) Informações ou outros dados não obtidos através desta perícia poderão ser obtidos junto à operadora de telefonia celular, mediante solicitação judicial.

Para garantir a preservação dos vestígios e permitir eventuais quesitos complementares, sugere-se que o equipamento periciado continue apreendido até o trânsito em julgado da eventual ação penal.

O referido material acompanha a primeira via do presente laudo, acondicionado em embalagem identificada, **lacrada com o nº 2222** e rubricada.

Este laudo foi redigido pelo Perito que o subscreve em primeiro lugar e impresso em 02 (duas) folhas de papel timbrado do Instituto. Acompanha o presente laudo 01 (um) **Anexo** impresso em 05 (cinco) folhas de papel timbrado deste Instituto. E são essas as declarações que em suas consciências têm os Peritos a fazer. E, por nada mais haver, deu-se por findo o exame solicitado que de tudo se lavrou o presente laudo que vai devidamente assinado pelos Peritos.

Luiz Rodrigo Grochocki
PERITO CRIMINAL

Raphael Laércio Zago
PERITO CRIMINAL

FLS. 1
ANEXO AO LAUDO Nº 222222-2

1. EQUIPAMENTO 1
1.1. Aparelho Celular:
1.1.1. Informações Gerais da Captura:

TABELA 1	
Parâmetro	Valor
Fabricante selecionado:	Nokia GSM
Modelo selecionado:	1200/1208
Fabricante detectado:	(Indisponível)
Modelo detectado:	(Indisponível)
Revisão:	(Indisponível)
IMEI:	356402028331070
Código de Segurança:	12345
Início da extração:	25/02/2014 11:31:04
Fim da extração:	25/02/2014 11:31:23
Data/Hora do telefone:	(Indisponível)
Tipo de conexão:	Serial Cable
Versão da UFED:	2.2.5.0 UFED; S/N:5534651
Extração suportada para:	Agenda Telefônica, SMS, Chamadas

1.1.2. Agenda Telefônica gravada na memória do aparelho, da forma como se apresentava por ocasião do exame:
Nenhuma informação foi retornada pelo equipamento forense "Cellebrite UFED".

1.1.3. Chamadas Realizadas gravadas na memória do aparelho, da forma como se apresentava por ocasião do exame:

TABELA 2					
Item	Chamada Realizada	Data	Hora	Duração	Identificação
1	95404318	19/02/2014	15:37:31	00:00:18	Sim A
2	95404318	19/02/2014	15:38:12	(*)	Sim A
3	95404318	19/02/2014	15:39:01	(*)	Sim A
4	95404324	19/02/2014	15:39:45	00:00:32	(nada consta)
5	95404324	19/02/2014	15:40:56	00:00:28	(nada consta)
6	95404318	19/02/2014	15:44:44	(*)	Sim A
7	95404318	19/02/2014	15:50:03	00:00:24	Sim A

(*) Informação não retornada pelo equipamento forense "Cellebrite UFED".

1.1.4. Chamadas Recebidas gravadas na memória do aparelho, da forma como se apresentava por ocasião do exame:

TABELA 3					
Item	Chamada Recebida	Data	Hora	Duração	Identificação
1	95404324	19/02/2014	15:43:53	00:00:36	(nada consta)
2	95404297	20/02/2014	10:16:28	00:00:10	Sim I
3	95404297	20/02/2014	10:18:38	00:00:17	Sim I

FLS. 2
ANEXO AO LAUDO Nº 222222-2

4	95404324	20/02/2014	11:32:28	00:00:20 (nada consta)
5	95404297	20/02/2014	11:37:25	00:00:15 Sim I
6	95404297	20/02/2014	11:37:51	00:00:16 Sim I
7	95404297	20/02/2014	11:38:17	00:00:28 Sim I

1.1.5. Chamadas Não Atendidas gravadas na memória do aparelho, da forma como se apresentava por ocasião do exame:
Nenhuma informação foi retornada pelo equipamento forense "Cellebrite UFED".

1.1.6. Mensagens de Texto (SMS) Recebidas gravadas na memória do aparelho, da forma como se apresentava por ocasião do exame:

TABELA 4				
Item	Mensagem Recebida	Data	Hora	Remetente
1	TIM RECADO: Ainda tem mensagem pra você! Ligue agora para *100 e escute seus recados.	10/12/2013	19:16:43	Tim
2	INFINITY RECADO: Voce recebeu um novo recado. Para ouvir ligue *100 ou responda com SIM para ler o recado no SMS. Veja os termos de uso em www.tim.com.br	10/12/2013	19:16:43	100
3	Ora José, posso ir amanha e quinta no CAIF? Boa noite! Julia!	10/12/2013	20:38:26	04188474138
4	BB informa: gerado codigo para liberacao de computador: FFGH, em 10/12/2013, as 21h23. Para sua seguranca, nao informe este codigo a ninguem.	10/12/2013	21:38:09	40040001
5	Oferta dupla! A TIM oferece ao 41991913251 um desconto exclusivo p/acesso a Internet! Atualize agora p/ o TIM Infinity Web + Torpedo p/ R$0,25 a mais/dia que usar! Envie "OK" p/5000 e ative a oferta; ganhe um premio surpresa garantido e entre no sorteio de R$100.000!	11/12/2013	14:49:52	5000
6	Urgente! Envie "ATIVARMODEM" p/1616 GRATIS pra ativar! Vc foi escolhido p/ Infinity Web Modem: 60MB diarios de internet pra navegar (so 1,99/dia q usar)!	12/12/2013	15:48:06	1616
7	INFINITY RECADO: Voce recebeu um novo recado. Para ouvir ligue *100 ou responda com SIM para ler o recado no SMS. Veja os termos de uso em www.tim.com.br	15/12/2013	15:15:04	100
8	TIM RECADO: Ainda tem mensagem pra você! Ligue agora para *100 e escute seus recados.	15/12/2013	15:15:04	Tim
9	Voce recebeu ligacoes de: <15/12> 04141998474B2 <3> 15:18hs <14/12> 04143992613314 <1> 18:43hs <13/12> 04141973473B6 <1> 19:09hs	15/12/2013	16:01:54	Te Ligou
10	Agora e de graca! Quando ligar para um numero TIM e ele nao puder atender, deixe um recado! E GRATIS! Aproveite mais esse beneficio da TIM pra voce!	16/12/2013	09:56:05	TIMDicas
11	Voce pode almocar comigo hoje? Beijos	18/12/2013	09:52:55	04199847482
12	Saio daqui às 13h e te mando uma mensagem	18/12/2013	09:57:07	04199847482
13	Estou saindo agora	18/12/2013	12:54:54	04199847482
14	To imprimindo	18/12/2013	16:55:46	04199847482

Figura 2 – Exemplos de Laudo Pericial Criminal

Para [Reis 2011] o laudo pericial criminal "é um dos itens mais importantes no estuda da Criminalística, pois é através dele que os exames são expressos e que a prova material do crime é manifestada". Deve apresentar "seu conteúdo exclusivamente voltado para o rigor das Leis Naturais, evitando qualquer relação com as Leis Jurídicas e com as Leis da Consciência".

Uma das tarefas mais árduas enfrentada diariamente pelos peritos é preparar o Laudo Pericial, isto é, expressar em papel, uma opinião formulada por ele relativo a um determinado caso. Esta opinião deve freqüentemente acomodar uma análise complexa. O laudo concluído estará sujeito à revisão crítica por via de testemunhos, conferências de pré-julgamento, e talvez um interrogatório rigoroso por vários advogados. Tem-se, portanto, que o Laudo Pericial é a expressão final do trabalho de perícia. No Sistema

SiCReT, o fluxo desde a origem dos dados a partir do dispositivo móvel até a elaboração do laudo pericial é mostrado na Figura 3.

Figura 3 – Fluxo: Do Dispositivo ao Laudo Pericial Criminal

Assim, foi adotado como padrão o XML, visto ser uma linguagem de marcação, recomendada pela W3C (*World Wide Web Consortium*) para a criação de documentos contendo dados. É expansível por não especificar as *tags* (conjunto de marcas), nem uma gramática. As *tags* definidas em um determinado arquivo só terão significado para um *parser* (analisador de marcas) específico para interpretar aquele tipo de dados (McLaughin 2000]. A linguagem XML pode ser usada para fornecer informações sobre a estrutura e o significado dos dados, ao invés de servir apenas como linguagem de marcação a exemplo do HTML (*Hypertext Markup Language*).

3. Banco de Dados

Esta seção apresenta o projeto de banco de dados do sistema SiCReT, tendo sido adotada a Orientação a Objetos utilizando a representação UML (*Unified Modeling Language*), modelo adotado por grande parte de sistemas orientados a objetos visto que constitui uma linguagem de modelagem não proprietária de terceira geração.

O OMG (*Object Management Group*) afirma que antes da codificação, a modelagem é uma parte essencial de projetos de software independentemente do tamanho da solução (http://www.omg.org/). Modelos ajudam a trabalhar em um alto nível de abstração e ao mesmo tempo facilitam o entendimento de uma solução. Para [Watson 2014] a representação UML auxilia a especificar, visualizar e documentar modelos de software, podendo ser utilizada para modelagem de negócios e de outros sistemas não-softwares. Ferramentas baseadas em UML possibilitam uma melhor compreensão das funcionalidades e facilitam a manutenibilidade do software. Portanto, o projeto do sistema SiCReT utilizou a modelagem UML para representar soluções que envolvem banco de dados, codificação JAVA e mineração de dados.

Visando atender as atividades dos peritos, o banco de dados considera as seguintes entidades: laudos, equipamentos, agenda, mensagens, chamadas, peritos, laudos_peritos, reus e arquivos, a partir do Dicionário de Dados mostrado na Tabela 2.

Tabela 2 - Dicionário de Dados do Modelo Relacional

TABELA	CAMPO	TIPO	DESCRIÇÃO
LAUDOS	LAUDO_ID	Int	Identificação do Laudo
	Codigo	VarChar(45)	Código do Laudo
	Descricao	VarChar(100)	Descrição do Laudo
	Inicio	Date	Data da criação do Laudo
	Conclusao	Date	Data da conclusão do Laudo
	Emissão	Date	Data da emissão do Laudo
	Situacao	VarChar(45)	Status de andamento do Laudo
	Oficio	VarChar(45)	Número do ofício
	Inquerito	VarChar(45)	Número do Inquérito
	Processo_Criminal	VarChar(45)	Número do Processo Criminal
EQUIPAMENTOS	EQUIPAMENTO_ID	Int	Identificação do Equipamento
	LAUDO_ID	Int	Identificação do Laudo
	Descricao	VarChar(100)	Descrição do Equipamento
	Fabricante	VarChar(45)	Nome do Frabricante
	Modelo	VarChar(45)	Modelo do Equipamento
	Fabricante_detectado	VarChar(45)	Nome do Frabricante Detectado
	Modelo_detectado	VarChar(45)	Modelo do Equipamento Detectado
	Revisao	VarChar(45)	Informações da Revisão do Equipamento
	Nome	VarChar(45)	Nome do Equipamento
	IMEI	VarChar(45)	International Mobile Equipment Identity
	NroSerie	VarChar(45)	Número de Série do Equipamento
	ICCID	VarChar(45)	International Circuit Card ID
	IMSI	VarChar(45)	International Mobile Subscriber Identity
	EndBluetooth	VarChar(45)	Endereço Bluetooth
	EndWiFi	VarChar(45)	Endereço Wi-Fi
	Inicio_Extracao	DateTime	Início da extração
	Fim_Extracao	DateTime	Fim da extração
	DataHora_telefone	VarChar(45)	Data/Hora do telefone
	Tipo_Conexao	VarChar(45)	Tipo de conexão
	Operadora	VarChar(45)	Operadora do Equipamento
	UFED	VarChar(45)	*Universal Forensic Extraction Device*
	VersaoUFED	VarChar(45)	Versão UFED
AGENDA	AGENDA_ID	Int	Identificação do Registro da Agenda
	LAUDO_ID	Int	Identificação do Laudo
	EQUIPAMENTO_ID	Int	Identificação do Equipamento
	Contato	VarChar(100)	Nome / Apelido do Contato
	Numero	VarChar(100)	Número do Contato
MENSAGENS	MENSAGEM_ID	Int	Identificação da Mensagem
	LAUDO_ID	Int	Identificação do Laudo
	EQUIPAMENTO_ID	Int	Identificação do Equipamento

	Item	Int	Numeração do Item
	Conteudo	LongText	Conteúdo da Mensagem
	Data	Date	Data de Criação da Mensagem
	Hora	Time	Hora de Criação da Mensagem
	NumeroDestino	VarChar(45)	Número do contato de Destino
	NumeroOrigem	VarChar(45)	Número do contato de Origem
	Situacao	VarChar(45)	Status: Enviada/Recebida/Rascunho
CHAMADAS	CHAMADA_ID	Int	Identificação da Chamada
	LAUDO_ID	Int	Identificação do Laudo
	EQUIPAMENTO_ID	Int	Identificação do Equipamento
	Item	Int	Numeração de itens por laudo
	Data	Date	Data da criação do registro
	Hora	Time	Hora da criação do registro
	NumeroDestino	VarChar(45)	Número do contato de Destino
	NumeroOrigem	VarChar(45)	Número do contato de Origem
	Situacao	VarChar(100)	Status da Chamada: Realizada, Recebida ou Não Atendida
	Tempo	Time	Tempo de Duração da Chamada
PERITOS	PERITO_ID	Int	Identificação do Perito
	Nome	VarChar(100)	Nome do Perito
	CPF	VarChar(20)	CPF do Perito
LAUDOS_ PERITOS	LAUDOS_PERITOS_ID	Int	Identificação do Registro da Tabela
	LAUDO_ID	Int	Identificação do Laudo
	PERITO_ID	Int	Identificação do Perito
	Inicio	Date	Data de início de participação do Perito
	Conclusao	Date	Conclusão da participação do Perito
	Situacao	VarChar(45)	Status do trabalho do Perito
REUS	PESSOA_ID	Int	Identificação do Registro da Tabela
	LAUDO_ID	Int	Identificação do Laudo
	Nome	VarChar(100)	Nome do Réu
	DctoIdetificacao	VarChar(45)	Documento de Identificação
	TipoDcto	VarChar(45)	Tipo de Documento de Identificação
ARQUIVOS	ARQUIVO_ID	Int	Identificação do Registro da Tabela
	LAUDO_ID	Int	Identificação do Laudo
	Arquivo	VarChar(200)	Nome e Localização do Arquivo
	Extensão	VarChar(10)	Extensão do Arquivo
	Tipo	VarChar(20)	Tipo: Laudo/Anexo/Complementares
	Hash	VarChar(200)	Código Hash do Arquivo

O modelo lógico de dados relativo ao BD Relacional (Figura 4) descreve as relações entre as tabelas do banco de dados, as chaves primárias, as chaves secundárias e os atributos de cada tabela.

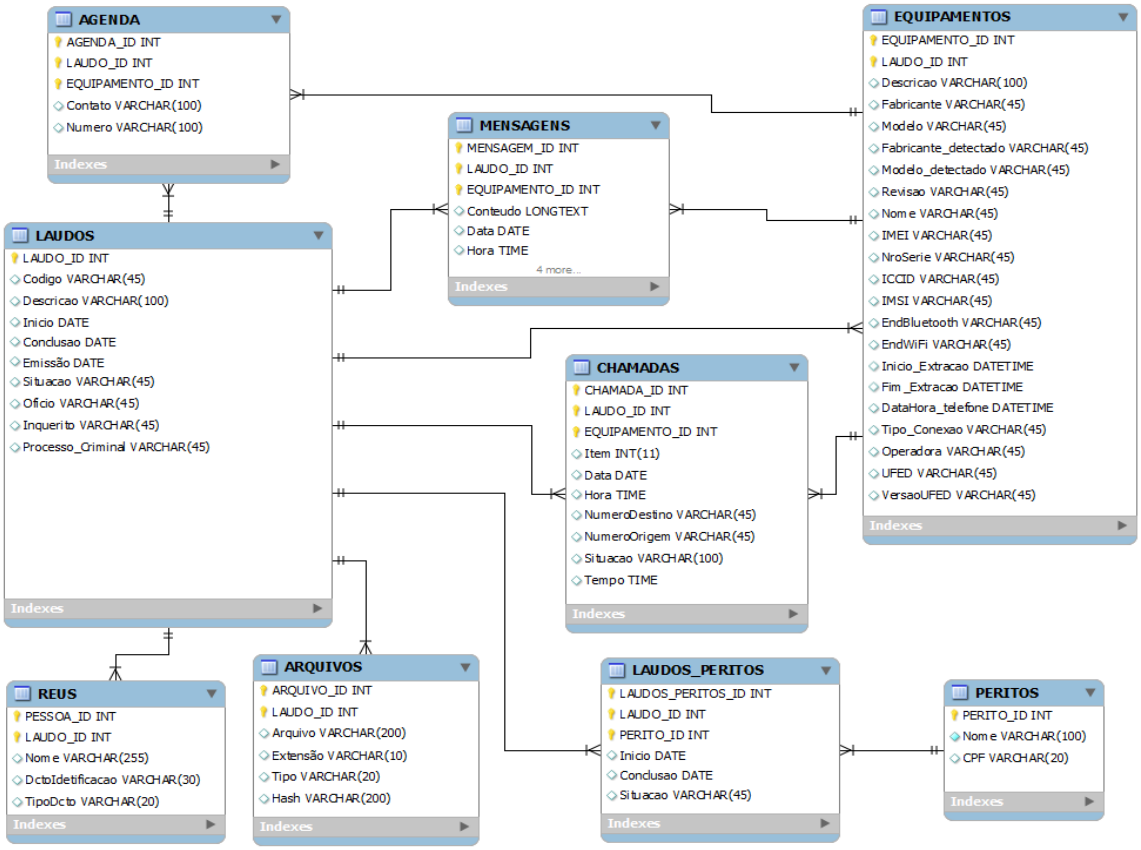


Figura 4 – Modelo Lógico de Dados

4. Estudo de Caso e Teste de Consistência do Banco de Dados

O Estudo de Caso por meio de uma situação simulada permite a criação de um ambiente controlado para prova de conceitos, com apoio dos peritos da Seção de Informática Forense do Instituto de Criminalística da Polícia Científica, da mesma forma que os peritos se deparam no exercício da profissão. Deste modo, o sistema poderá detectar, identificar, analisar e produzir relatórios e grafos demonstrando interseções entre as informações contidas nos laudos de dispositivos móveis armazenados no banco de dados.

As informações consideradas nesse momento, para a instanciação do Banco de Dados e realização de testes de consistência, consideram a extração de dados a partir de 10 (dez) cartões SIM, com tecnologia GSM, de 128KB, estabelecendo uma situação simulada (Figura 5).

Para instanciar a base de dados e realizar testes de validação não se pode, em situações que envolvam crimes reais, utilizar laudos reais já elaborados pelos peritos, visto que não se pode recuperar junto ao Instituto de Criminalística um conjunto de laudos que tenham relações entre si de modo a representar situação semelhante à situação desejada. A situação, então, simulada possui 4 grupos de dispositivos, com grupos entre $n_{max} = 5$ e $n_{min} = 1$, sendo n o número de dispositivos de cada grupo.

Além disto, os grupos 1, 2 e 3 relacionam-se entre si por meio dos dispositivos SIM B, SIM F e SIM I. Como já apresentado anteriormente, são de interesse os seguintes dados:

- Agenda: Contatos Telefônicos;
- Chamadas: Realizadas, Recebidas e não-Atendidas;
- Mensagens de Texto (SMS): Mensagens de Texto (SMS): Recebidas, Enviadas, Rascunhos, não-Enviadas (caixa de saída).

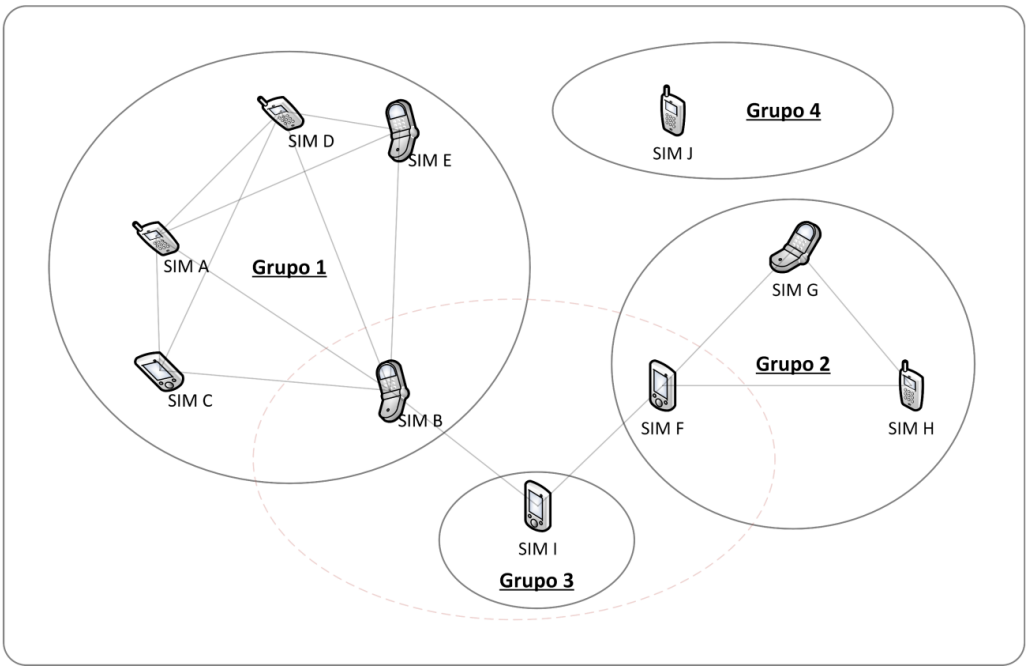


Figura 5 - Representação Gráfica de Cruzamentos entre Dispositivos Móveis

O Estudo de Caso, então, utiliza alguns Tipos de Operações que são responsáveis por gerar os registros necessários nos dispositivos, fornecendo assim dados suficientes para a realização dos testes, de acordo com os peritos do Instituto de Criminalística. Assim, as operações de interesse podem ser listadas da seguinte maneira:

1) Registro de Contato Telefônico;
2) Chamada Realizada para Contato Registrado;
3) Chamada Realizada para um Nº Telefônico;
4) Chamada Recebida de um Contato Registrado;
5) Chamada Recebida de um Nº Telefônico;
6) Chamada Não Atendida de um Contato Registrado;
7) Chamada Não Atendida de um Nº Telefônico;
8) SMS recebido de um Contato Registrado;
9) SMS recebido de um Nº Telefônico;
10) SMS enviado para um Contato Registrado;
11) SMS enviado para um Nº Telefônico;
12) SMS armazenado nos rascunhos de um Contato Registrado;
13) SMS armazenado nos rascunhos de um Nº Telefônico;
14) SMS não enviadas de um Contato Registrado;
15) SMS não enviadas de um Nº Telefônico.

Estão sendo utilizados ainda cinco números externos, que não apresentam uma ligação direta com os equipamentos mencionados na situação simulada, representando apenas um Número de Telefone Externo que em algum momento é acionado por um contato, uma ligação ou uma mensagem de texto. As relações entre os dispositivos móveis da situação simulada, considerando-se os 3 objetos: agenda, chamadas e mensagens; perfazem um entrelaçamento de dados, de modo a permitir que a extração dos dados dos dispositivos, a elaboração dos laudos e a formação do banco de dados represente fielmente a situação simulada (Figura 6). Não se encontram representadas os tipos de operações (se a mensagem foi enviada ou recebida, por exemplo), mas somente a existência do objeto.

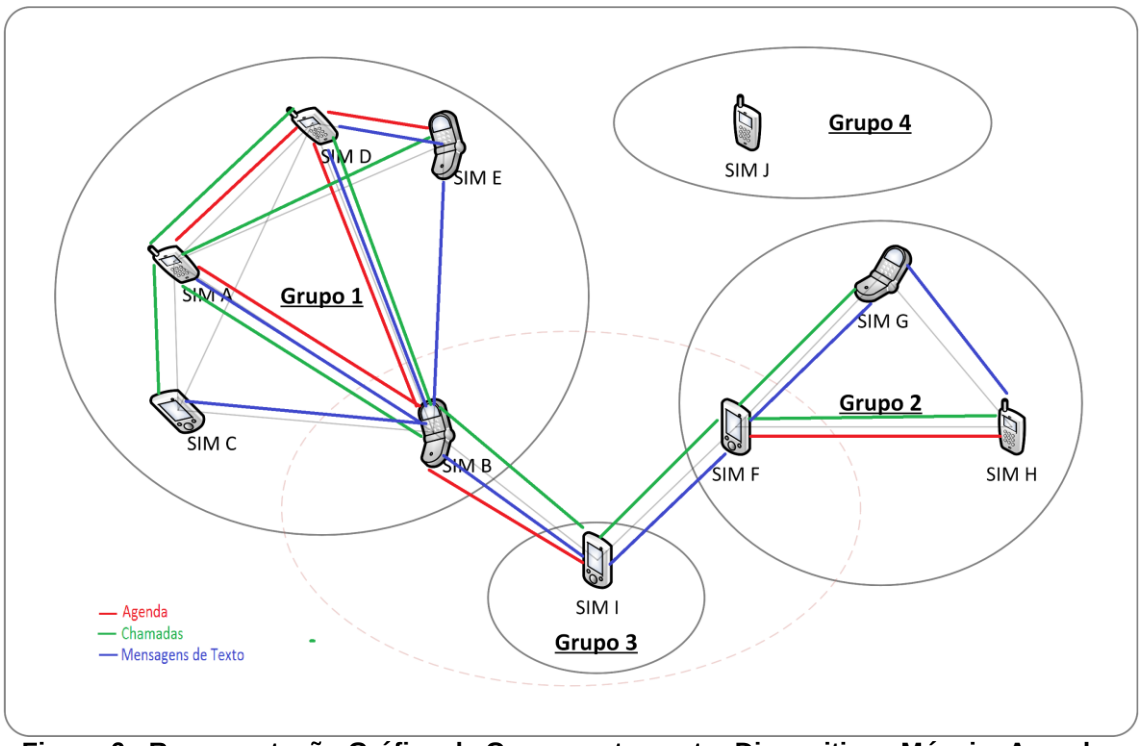


Figura 6 - Representação Gráfica de Cruzamentos entre Dispositivos Móveis: Agenda, Chamadas e Mensagens de Texto.

5. Protótipo do Sistema SiCReT

O sistema de informações SiCReT tem como padrão de desenvolvimento o modelo MVC (*Model-View-Control*) que garante a separação da interface, do controle de fluxo e das regras do projeto [Hemrajani 2007], sendo que cada tipo de componente executa um determinado tipo de tarefa [Moreira Neto 2009]. Assim, é possível realizar alterações em cada uma das camadas isoladamente, diminuindo o retrabalho proveniente da necessidade de atualização em aplicações Web. A Tabela 3 mostra as ferramentas e tecnologias utilizadas, até o presente momento, no desenvolvimento do sistema SiCReT.

A segurança de um sistema desenvolvido utilizando a plataforma Java conta com um conjunto de pacotes disponibilizados por frameworks específicos, dentre eles citamos o JSSE (*Java Secure Socket Extension*) e o JAAS *(Java Authentication and Authorization Service)* [Garms e Somerfield 2001].

Tabela 3 - Ferramentas e Tecnologias

NOME	DESCRIÇÃO
Eclipse Platform - Indigo 3.7.1	Criação do projeto, programação Java, serviços servidor Tomcat. Disponível em: http://www.eclipse.org/
PRIMEFACES 3.0	Framework de componentes ricos. Disponível em: http://www.primefaces.org/
Hibernate 3.5.6	Persistência dos informações no Banco de Dados. Disponível em: https://www.hibernate.org/
Spring 3.0.5	Framework de Inversão de Controle e Injeção de Dependência. Disponível em: http://www.springsource.org/
MySQL 5.5.25	Sistema Gerenciador de Banco de Dados. Disponível em: http://www.mysql.com/
Apache Tomcat 7.0.21	Contêiner de aplicações web. Disponível em: http://tomcat.apache.org/
MySQL Workbench 5.2.34 CE	Interface gráfica para administrar e trabalhar com o banco de dados MySQL. Disponível em: http://www.mysql.com/
JSSE	Java Secure Sockets Extension
JAAS	Java Authentication an Autorization Service

O JSSE provê suporte ao SSL *(Secure Sockets Layer*) e TLS (*Transport Layer Security*) permitindo que a comunicação entre Servidores Web e seus respectivos clientes seja realizada pelo protocolo HTTPS (*HiperText Transfer Protocol Secure*), ou seja, sobre uma conexão segura. Utilizando métodos de criptografia que garantem tanto a integridade quanto a confidencialidade das informações que trafegam na rede, além de permitir o uso de certificados digitais para a autenticação das partes envolvidas.

Além de contar com a segurança da infra-estrutura de rede, do sistema servidor e do SGBD o sistema de informações SiCReT conta ainda com as funcionalidades de gerenciamento de usuários disponibilizadas pelo JAAS, framework que permite as aplicações Java realizarem a identificação, autenticação e controle de acesso ao nível de usuário nos recursos disponibilizados pelo sistema.

A estratégia de segurança imposta pelo Java é aumentada pelos recursos de segurança do sistema operacional no qual está sendo executada e reforçada pelo SGBD. Uma aplicação Java com uma estratégia de segurança pode tentar ler um arquivo em específico, mas se o usuário que está executando a aplicação não tiver permissão para o determinado arquivo, a aplicação Java não terá êxito [Oaks 1999].

Sob a ótica da segurança computacional, o sistema será auditável, ou seja, controlará o acesso e a inserção de informações por meio de usuários, de modo a registrar as operações realizadas pelos usuários, permitindo estabelecer quem realizou o quê no sistema.

6. Conclusão

Este artigo propôs um Banco de Dados de laudos periciais em dispositivos móveis, apresentado o modelo lógico dos dados e o dicionário de dados. Além disto, por meio de uma situação simulada, descreve-se um estudo de caso que está permitindo a

instanciação do BD, bem como, a realização de testes de validação e provas de conceitos. Tal BD integra um sistema computacional de maior proporção e permitirá a aplicação de técnicas de mineração de dados [Uthra, Tech 2014]. É importante observar que a criação e disponibilização deste BD permitirão a integração entre os diversos Institutos de Criminalística do Brasil, auxiliando no entendimento sobre as forças que governam o fluxo criminoso e disponibilizando ferramentas de apoio aos Serviços de Inteligência e Policiamento Preditivo. Além disto, outras pesquisas poderão ser realizadas à medida que a situação simulada for ampliada de acordo com as necessidades práticas dos peritos diante de casos e situações reais.

7. Referências

Garms, J.; Somerfield, D.. "Professional Java Security". Wrox Press Ltd, 2001.

Grochocki, L. R.; Vrubel, A.; Zago, R. L.; Decarli, A.; Freitas, C. O. A.. SiCReT - Sistema de Cruzamento de registros Telefônicos. In: XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2013, Manaus: Sociedade Brasileira de Computação, 2013. v. 1. p. 527-536.

Hemrajani, A. "Desenvolvimento Ágil em JAVA com SPRING HIBERNATE E ECLIPSE". Pearson Education, 2007.

Jansen, W.; Ayers, R. "Computer Security - guidelines on cell phone forensics". National Institute of Standards and Technology – NIST, Special Publication 800-101, May 2007, 104 p. Disponível em <http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf> Acesso em maio 2014.

McLaughlin, B. "Java and XML". Mike Loukides, 2000.

MOREIRA NETO, O. "Entendendo e Dominando o Java para Internet". Digerati Books, 2009. 320p.

OAKS, S. "Segurança de dados em Java". Editora Ciência Moderna, 1999.

REIS, A. B. "Metodologia científica em perícia criminal". Campinas, SP: Millenium, 2011.

ROSA, M. V. F. "Perícia Judicial - Teoria e Prática". Sérgio Antônio Fabris Editor, 1999.

Ultra, R. G.; Tech, M. "Data Mining Techniques to Analyze Crime Data". International Journal For Technological Research In Engineering, Volume 1, Issue 9, May-2014, p. 882-884.

Watson, A. "Visual Modelling: past, present and future". Disponível em <http://www.uml.org/Visual_Modeling.pdf> Acesso em maio de 2014.

WEISER, M. "Some Computer Science Issues in Ubiquitous Computing". Communications of the ACM, v. 265, n. 3, 1993, p. 137 - 143.

NPDI Find Porn: Uma Ferramenta para Detecção de Conteúdo Pornográfico

Ramon F. Pessoa[1], Edemir Ferreira de A. Junior[1], Carlos A. Caetano Junior[1], Silvio Jamil F. Guimarães[2], Jefersson A. dos Santos[1], Arnaldo de A. Araújo[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos 6627 - Prédio do ICEx - Pampulha
31270-010 - Belo Horizonte - MG - Brasil
[2]Pontifícia Universidade Católica de Minas Gerais, VIPLAB - ICEI/PUC Minas, Minas Gerais, Brasil

{ramon.pessoa,edemirm,carlos.caetano,jefersson,arnaldo}@dcc.ufmg.br

sjamil@pucminas.br

Abstract. *With the growing amount of content deemed inappropriate on the Internet, such as pornography, the need for this type of material filter came up. The reason for this is given by the fact that such content is often banned in certain environments (e.g., workplaces and schools), and especially when dealing with child pornography, considered one of the most reported virtual crimes in Brazil according to the site Safernet[1]. In recent years, many works of literature have been mainly focused on the detection of pornographic images and videos. This work presents a tool to detect pornographic content created from recent academic work developed by NPDI[2] research group.*

Resumo. *Com o crescimento da quantidade de conteúdos considerados inapropriados na Internet, como pornografia, surgiu a necessidade de filtros para tal tipo de material. O motivo é dado pelo fato de que esse tipo de conteúdo é frequentemente proibido em certos ambientes (como, locais de trabalho e escolas), e principalmente se tratando de pornografia infantil, considerado um dos crimes virtuais mais denunciados no Brasil de acordo com o site Safernet[1]. Nos últimos anos, diversos trabalhos da literatura têm tido como foco principal a detecção de imagens e vídeos pornográficos. Neste trabalho, é apresentada uma ferramenta de detecção de conteúdos pornográficos, criada a partir dos recentes trabalhos acadêmicos desenvolvidos pelo grupo de pesquisa NPDI[2].*

1. Introdução

Com o advento da tecnologia, em especial o aumento do acesso à Internet, originou-se nos últimos anos uma grande quantidade de informação disponível ao público, como vídeos e imagens. Nesse contexto, algumas situações exigem que haja um controle em relação ao conteúdo destes materiais. Em especial, apresenta-se a detecção de pornografia como uns dos grandes desafios atuais.

Detectar e filtrar conteúdo visual pornográfico é uma preocupação em vários ambientes como, escolas, empresas, igrejas e outros locais públicos. Uma maneira para se

[1]Safernet - www.safernet.org.br (03/Set/2014).
[2]Núcleo de Processamento Digital de Imagens (NPDI) - www.npdi.dcc.ufmg.br.

realizar tal tarefa é a utilização de palavras chaves textuais vinculadas a informações multimídia, porém [Lopes 2009] mostrou que tal abordagem não é suficiente para detecção deste tipo de conteúdo .

Nos últimos anos, diversos trabalhos da literatura têm tido como foco principal a detecção de imagens e vídeos pornográficos baseados em conteúdo visual como alternativa ao uso exclusivo de informações textuais [Steel 2012], [Avila et al. 2013], [Yu and Han 2014], [Caetano et al. 2014a], [Caetano et al. 2014b].

Abordagens baseadas em características locais, em conjunto com modelos *Bag-of-Words* (BoW), têm sido aplicadas com sucesso em tarefas de classificação para reconhecimento de padrões visuais [Agarwal et al. 2004], [Yang et al. 2007]. Neste tipo de abordagem, a maior vantagem é a não necessidade de um modelo explícito do objeto, dado que a diversidade de características da imagem (como forma, escala ou iluminação) é tratada por um conjunto de treinamento que representa essa variabilidade. Portanto, a utilização do modelo BoW se torna uma abordagem interessante no contexto de detecção de pornografia.

Nesse artigo, apresenta-se uma nova ferramenta para detecção de conteúdo pornográfico em imagens, chamada NPDI *Find Porn*. Foram utilizadas metodologias recentes para o desenvolvimento de uma ferramenta prática e intuitiva para detecção de imagens com conteúdo pornográfico. O objetivo é fornecer suporte a usuários que não possuem conhecimento suficiente de técnicas computacionais para reconhecimento visual. A ferramenta é baseada no método proposto por [Caetano et al. 2014a] que utiliza descritores binários em conjunto com uma extensão do modelo BoW, proposto por [Avila et al. 2013], se diferenciando na etapa de classificação onde foi utilizado o classificador *Gradient Tree Boosting* [Friedman 2002] para predizer se uma imagem é pornográfica ou não.

Indivíduos e profissionais que podem precisar deste sistema de detecção de pornografia, são listados abaixo:

1. Pais: Os pais podem usar o NPDI *Find Porn* para reduzir as chances das crianças se depararem com pornografia presente no computador em que elas estejam usando.
2. Empregados: As empresas podem usar o NPDI *Find Porn* para remover conteúdo pornográfico em computadores que expõe a empresa a riscos legais.
3. Escolas e igrejas: Escolas e igrejas podem evitar imagens pornográficas indesejadas de seus computadores usando esta ferramenta. Apesar dos filtros disponíveis na Internet, conteúdos pornográficos podem facilmente contaminar os computadores de uma organização. Estes conteúdos são copiados de *pen drives* ou outras mídias, baixados a partir de e-mail, ou simplesmente perdidos por seu filtro.
4. Profissionais de perícias, profissionais da lei, policiais, entre outros: Profissionais envolvidos na aplicação da lei sabem que se perde muito tempo ao se procurar evidências digitais em laboratórios judiciais. O NPDI *Find Porn* é uma ferramenta de análise para determinar se imagens pornográficas estão presentes em um computador sem treinar e envolver examinadores judiciais. Essa ferramenta pode ser incorporada em dispositivo móvel não precisando de dependências instaladas no sistema operacional.

O restante desse artigo está organizado em quatro seções. A Seção 2 apresenta um breve resumo dos conceitos necessários para o entendimento do processo de reconhecimento de padrões visuais. A Seção 3 descreve o funcionamento do método implementado no *software*. A Seção 4 discute brevemente como utilizar o *software* proposto. E, finalmente, a Seção 5 conclui esse artigo direcionando trabalhos futuros.

2. Conceitos

Segundo [Chatfield et al. 2011], a abordagem de reconhecimento de padrões visuais mais utilizada na literatura pode ser dividida em três etapas distintas: (i) extração de características locais da imagem; (ii) codificação das características locais em uma representação intermediária (*mid-level*); e (iii) classificação da representação intermediária, geralmente, baseada em técnicas de aprendizado de máquina.

O *software* implementado utiliza uma abordagem baseada nessas três etapas. Cada uma delas é detalhada a seguir.

2.1. Extração de Características Locais

Segundo [Tuytelaars and Mikolajczyk 2008], características locais consistem em padrões de imagem que se diferem de sua vizinhança, geralmente, associados às mudanças nas propriedades da imagem (textura e contraste por exemplo). A extração de características locais é a primeira etapa a ser feita em um processo que envolva reconhecimento de padrões visuais. Uma maneira de se realizar tal etapa consiste em selecionar *patches* da imagem que contenham informações relevantes, e então descrevê-los com o uso de algum descritor de características.

De acordo com [Tuytelaars 2010], a seleção dos *patches* pode ser feita com base em dois tipos de abordagens: (i) utilizando pontos de interesse, neste caso é aplicado um algoritmo para encontrar tal região a ser descrita; ou (ii) amostragem densa, onde regiões de tamanho fixo são alocadas em uma grade de tamanho regular. A Figura 1 ilustra um exemplo de extração de características locais com cada abordagem.

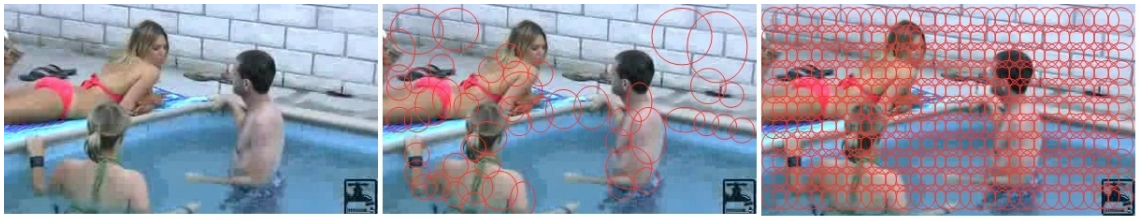


Figure 1. Exemplo de características locais extraídas pelas abordagens de pontos de interesses e amostragem densa. Cada círculo vermelho representa uma característica local a ser extraída.

Um descritor de características pode ser considerado como uma função aplicada em uma região de uma imagem com o objetivo de descrevê-la. Uma maneira bem simples de se descrever uma região seria representar todos os *pixels* desta região em um único vetor. No entanto, dependendo do tamanho da região a ser descrita, isso resultaria em um vetor de alta dimensionalidade, levando também a uma alta complexidade computacional para um futuro reconhecimento desta região [Mikolajczyk and Schmid 2005].

Os vetores gerados pelos descritores de características mais comuns na literatura são compostos por valores reais, que são calculados utilizando uma técnica baseada na contagem das ocorrências de orientações de gradiente nas regiões de uma imagem, como: SIFT *(Scale-Invariant Feature Transform)* [Lowe 2004], HOG (*Histograms of Oriented Gradients*) [Dalal and Triggs 2005] e SURF (*Speeded Up Robust Features*) [Bay et al. 2006].

Como uma alternativa de baixa complexidade, os descritores binários têm emergido recentemente [Canclini et al. 2013]. Este tipo de descritor tem recebido uma atenção considerável por gerar resultados similares, em alguns casos melhores, quando comparados a descritores não-binários do estado da arte. A ideia básica por trás dos descritores binários é poder codificar a maioria das informações de um *patch* em uma sequência binária, usando apenas simples testes binários comparando a intensidade entre os *pixels*. Isso pode ser feito de maneira bem rápida, já que apenas comparações de intensidade precisam ser calculadas.

2.2. Representação Intermediária

Com as características locais já extraídas, torna-se necessário codificá-las para que se tenha uma representação global da imagem. Uma maneira de se fazer isso é realizar uma quantização dessas características utilizando o modelo *Bag-of-Words* (BoW).

Segundo [Boureau et al. 2010], o modelo BoW pode ser compreendido como a aplicação de duas etapas críticas: codificação e *pooling*. A etapa de codificação quantifica as características locais extraídas da imagem de acordo com um dicionário visual, conhecido como *codebook*, associando os descritores locais extraídos da imagem com o elemento mais próximo deste vocabulário visual. O dicionário visual, normalmente, é construído aplicando um algoritmo de clusterização, geralmente *k-means* [Lloyd 1982], em um conjunto de amostras dos descritores locais extraídos, onde cada palavra visual (*codewords*) corresponde ao centroide obtido de cada *cluster*. A etapa de *pooling* resume as palavras visuais obtidas em um único vetor de características com o objetivo de representar toda a imagem.

A Figura 2 ilustra o processo de codificação e *pooling* descrito anteriormente.

Como uma extensão do modelo BoW, a representação intermediária BossaNova [Avila et al. 2013] oferece um aprimoramento na etapa de *pooling*, a fim de preservar de uma maneira mais rica a informação obtida durante a etapa de codificação. Desta maneira, em vez de compactar toda a informação relacionada a uma palavra visual em um único valor escalar, a etapa de *pooling* resulta em uma distribuição de distâncias. Para isto, [Avila et al. 2013] usaram uma estimação não-paramétrica da distribuição dos descritores, calculando um histograma de distâncias entre os descritores encontrados na imagem e cada palavra visual presente no dicionário visual.

[Avila et al. 2013] aplicaram a representação BossaNova no contexto de reconhecimento de objetos e detecção de pornografia. Em comparação ao modelo BoW, BossaNova se sobressai de maneira significante [Avila et al. 2011, Avila et al. 2012, Avila et al. 2013], apenas usando um simples histograma de distâncias para capturar as informações relevantes. BossaNova mostra ser um método muito flexível, mantendo uma representação final bem compacta.

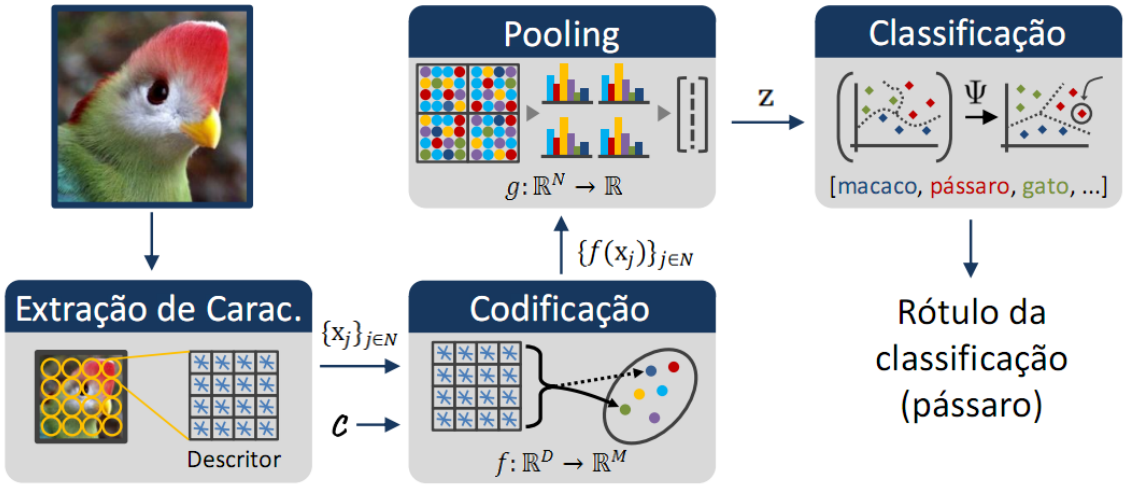


Figure 2. Processo de classificação usado pelo modelo *Bag-of-Words*. Primeiro, os descritores locais são extraídos da imagem. Na fase de codificação, uma função f ativa a palavra visual mais próxima ao descritor local, atribuindo peso zero a todas as outras. Em seguida, na etapa de *pooling*, a função g resume as palavras visuais obtidas em uma único vetor de característica z. Por fim, um algoritmo de classificação (por exemplo, *Support Vector Machine* [Cortes and Vapnik 1995]) é treinado com base nos vetores BoW obtidos. Imagem adaptada de [Chatfield et al. 2011].

2.3. Classificação Supervisionada

Segundo [Ghahramani 2004], aprendizado de máquina é o campo de pesquisa dedicado ao estudo formal de sistemas de aprendizagem. Pode ser considerado como um campo altamente interdisciplinar por se basear em ideias de diversas áreas, como estatística, ciência da computação, engenharia, ciência cognitiva, teoria de otimização, entre outras.

De acordo com [Dietterich 1997], o objetivo do aprendizado de máquina é construir modelos computacionais que podem adaptar-se e aprender a partir da experiência. Os algoritmos de aprendizado de máquina têm como objetivo descobrir o relacionamento entre as variáveis de um sistema (entrada/saída) a partir de dados amostrados anteriormente.

As técnicas de aprendizado de máquina podem ser separadas em várias categorias (supervisionado, não-supervisionado, semi-supervisionado, ativo, meta aprendizado), porém, de uma forma geral, a distinção mais fundamental é entre algoritmos de aprendizado supervisionado e não-supervisionado.

No aprendizado supervisionado, o algoritmo recebe como entrada uma quantidade de amostras com os seus respectivos rótulos, que serão utilizadas para que o algoritmo aprenda a distribuição de probabilidades daquela tarefa em específico (conjunto de treinamento). Logo depois, é oferecida uma quantidade de amostras sem os seus rótulos para que o algoritmo tente inferir os rótulos em função do que foi aprendido anteriormente (conjunto de teste).

Existem diversas estratégias direcionadas à tarefa de classificação. Dentre elas, são destacados os métodos *ensembles*, que podem ser divididos em dois conjuntos:

averaging e *boosting*. Os métodos *averaging* utilizam do princípio da construção de vários classificadores para, então, efetuar o cálculo da média de suas predições. Usualmente, essa abordagem apresenta resultados melhores do que a estimativa com apenas um classificador, devido à redução da variância do classificador final. Em contraste, os métodos *boosting* utilizam classificadores simples, para construir de forma iterativa um classificador final mais robusto, com baixo *bias*. Exemplos de métodos baseados em *averaging* seriam *Random Forest* [Breiman 2001], *Bootstrap Aggregating* [Breiman 1996] e *Extra-Trees* [Geurts et al. 2006]; enquanto *boosting*, tem-se *AdaBoost* [Freund and Schapire 1995], *Gradient Tree Boosting* [Friedman 2002], etc.

3. Metodologia

A Figura 3 ilustra o fluxograma da abordagem de Detecção de Imagens Pornográficas utilizado pelo sistema NPDI *Find Porn*. Esta abordagem é uma adaptação do método utilizado em [Caetano et al. 2014a] para imagens.

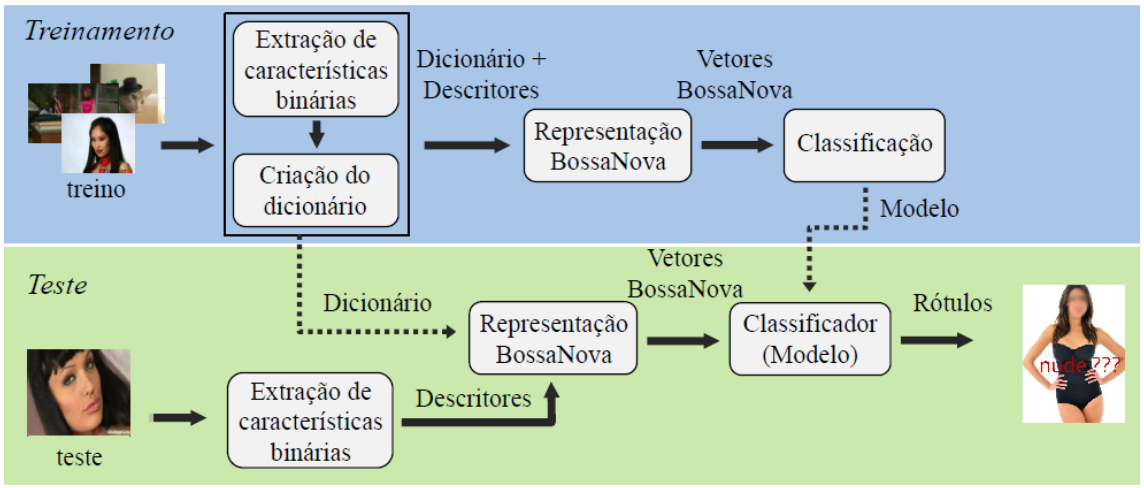


Figure 3. Fluxograma da abordagem de Detecção de Imagens Pornográficas [Caetano et al. 2014b] (Adaptado).

Primeiramente, são extraídas as características de cada imagem utilizando os descritores binários usando uma abordagem de amostragem densa. Em seguida, na fase de treinamento, é gerado o dicionário visual utilizando uma amostragem das características extraídas anteriormente. Gerado o dicionário visual, as características das imagens são codificadas para uma representação intermediária, que é transferida para a etapa de treinamento do classificador. Na etapa de classificação, o classificador recebe a representação intermediária da imagem requisitada e retorna um rótulo. Cada um dos passos citados são detalhados a seguir.

3.1. Extração de Características Locais utilizando Descritores Binários

O alto custo computacional dos descritores locais [Caetano et al. 2014a] inviabiliza a utilização dos mesmos na ferramenta desenvolvida. Segundo [Caetano et al. 2014b], a utilização de descritores binários para a detecção de conteúdo pornográfico apresenta resultados comparáveis a descritores não binários do estado da arte, além de apresentar custo computacional reduzido. No estudo, foram utilizados os descritores binários mais comuns na literatura:

1. BRIEF (*Binary Robust Independent Elementary Features*) [Calonder et al. 2010];
2. ORB (*Oriented Fast and Rotated Brief*) [Rublee et al. 2011];
3. BRISK (*Binary Robust Invariant Scalable Keypoints*) [Leutenegger et al. 2011];
4. FREAK (*Fast REtinA Keypoint*) [Alahi et al. 2012];
5. *BinBoost* [Trzcinski et al. 2013].

3.2. Dicionário Visual

Para a construção do dicionário visual, foi utilizado o método de agrupamento *k-medians* [Jain and Dubes 1988], que de acordo com [Caetano et al. 2014b], produz resultados melhores quando utilizados com descritores binários para a extração de características.

3.3. *Mid Level Representation*

Dado os resultados apresentados na literatura e nos recentes trabalhos ([Caetano et al. 2014a], [Avila et al. 2013]), foi utilizada a representação intermediária BossaNova combinada com descritores binários para uma codificação com maior representação das características extraídas.

3.4. Classificação

Para a etapa de classificação, foi utilizado um método chamado *Gradient Tree Boosting* (GTB) proposto por [Friedman 2002]. Assim como outros métodos de *boosting*, o GTB utiliza a combinação de classificadores fracos de maneira iterativa, para criar um classificador mais robusto.

Na fase de treinamento, é fornecido para o GTB um conjunto de amostras com as suas respectivas classes para que o método possa aprender a distribuição de probabilidade do cenário. Na fase de teste, é fornecida uma imagem ao classificador que gerará como saída uma premeditação para a imagem (pornográfica ou não pornográfica).

4. Utilização da Ferramenta

O desenvolvimento da ferramenta NPDI *Find Porn* é resultado das pesquisas recentes na área de detecção de pornografia do Laboratório NPDI. O objetivo era a criação de uma ferramenta prática e intuitiva, principalmente para usuários que não possuem o conhecimento teórico do procedimento de reconhecimento de padrões visuais.

Assim, a ferramenta foi desenvolvida para funcionar em um dispositivo de armazenamento móvel (como *pen drive*) que irá realizar uma pesquisa em todo o computador alvo procurando por imagens que serão avaliadas, utilizando o *framework* descrito anteriormente, como sendo imagens de conteúdo pornográfico ou não.

Uma das características principais do sistema é a não necessidade de qualquer tipo de instalação no sistema operacional, pois todas as dependências já se encontram dentro do *pen drive* contendo a ferramenta NPDI *Find Porn*, deixando assim a facilidade para que o usuário possa levá-lo a qualquer lugar e usá-lo em qualquer computador que possua.

Para a utilização da ferramenta, é necessário seguir os seguintes passos:

1. Logar em um computador;
2. Inserir o *Pen Drive* NPDI *Find Porn* no computador;
3. Executar o sistema NPDI *Find Porn*;

4. Selecionar o diretório do computador onde deseja fazer a pesquisa e iniciar a pesquisa;
5. Visualizar os resultados da pesquisa.

Estes passos são detalhados na Figura 4. A Figura 5 exibe a tela inicial da ferramenta NPDI *Find Porn* (Passo 1, 2 e 3) e Figura 6 mostra a ferramenta em execução (Passos 4 e 5). Uma apresentação do funcionamento da ferramenta desenvolvida pode ser encontrada no vídeo de demonstração do NPDI *Find Porn* [3].

5. Conclusão e Trabalhos Futuros

Neste trabalho, apresentou-se uma ferramenta de detecção de conteúdos pornográficos criada a partir dos recentes trabalhos acadêmicos desenvolvidos no laboratório NPDI da Universidade Federal de Minas Gerais (UFMG), com participação do VIPLab (Audio-Visual Information Processing Lab)[4] da Pontifícia Universidade Católica de Minas Gerais (PUC Minas).

O NPDI *Find Porn* é um sistema que irá vasculhar por todas as imagens em seu computador, procurando por conteúdo pornográfico, onde no final da pesquisa o sistema criará um relatório de imagens pornográficas suspeitas. O *software* de detecção de pornografia pode ser incorporado em um dispositivo móvel, sendo assim facilmente levado a qualquer lugar e ser usado em qualquer computador.

Entre as vantagens da ferramenta de detecção de conteúdo pornográfico, tem-se que o sistema permite proteger seu computador de pornografia indesejada. Muitos *sites* pornográficos contêm vírus que podem fazer muitos danos ao computador. Nestes casos, o NPDI *Find Porn* varre o computador oferecendo a segurança de que o seu computador esteja livre de imagens indesejadas ou mesmo ilegais. Outra vantagem, é que o sistema NPDI *Find Porn* evita que uma pessoa precise manualmente vasculhar um computador procurando por imagens pornográficas.

Dentre os trabalhos futuros, tem-se a adaptação do sistema NPDI *Find Porn* para detectar vídeos com conteúdos pornográficos, onde um algoritmo de extração de *frames* relevantes será desenvolvido, a extensão do sistema para outras plataformas (Linux e Mac), e será executado um protocolo de testes exaustivos para a ferramenta NPDI *Find Porn*.

6. Agradecimentos

Os autores gostariam de agradecer ao CNPq, à CAPES e à FAPEMIG pelo suporte financeiro recebido.

References

[Agarwal et al. 2004] Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1475–1490.

[3] Vídeo de demonstração da ferramenta NPDI *Find Porn* (Acessado em 03/Set/2014): `https://www.youtube.com/watch?v=ZNDfsxGHRDE&feature=youtu.be`

[4] Audio-Visual Information Processing Lab (VIPLab) - www.icei.pucminas.br/projetos/viplab

[Alahi et al. 2012] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. Ieee.

[Avila et al. 2012] Avila, S., Thome, N., Cord, M., Valle, E., and Araújo, A. d. A. (2012). Bossanova at imageclef 2012 flickr photo annotation task. *Working Notes of the CLEF*.

[Avila et al. 2011] Avila, S., Thome, N., Cord, M., Valle, E., and de A Araujo, A. (2011). Bossa: Extended bow formalism for image classification. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2909–2912. IEEE.

[Avila et al. 2013] Avila, S., Thome, N., Cord, M., Valle, E., and De A AraúJo, A. (2013). Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465.

[Bay et al. 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer.

[Boureau et al. 2010] Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566. IEEE.

[Breiman 1996] Breiman, L. (1996). Bias, variance, and arcing classifiers.

[Breiman 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Caetano et al. 2014a] Caetano, C., Avila, S., Guimarães, S., and Araújo, A. d. A. (2014a). Representing local binary descriptors with bossanova for visual recognition. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 49–54, New York, NY, USA. ACM.

[Caetano et al. 2014b] Caetano, C., Avila, S., Guimaraes, S., and Araújo, A. d. A. (2014b). Pornography detection using bossanova video descriptor. In *European Signal Processing Conference (EUSIPCO 2014)*, Lisbon, Portugal.

[Calonder et al. 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer.

[Canclini et al. 2013] Canclini, A., Cesana, M., Redondi, A., Tagliasacchi, M., Ascenso, J., and Cilla, R. (2013). Evaluation of low-complexity visual feature detectors and descriptors. In *Digital Signal Processing (DSP), 2013 18th International Conference on*, pages 1–7. IEEE.

[Chatfield et al. 2011] Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods.

[Cortes and Vapnik 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[Dalal and Triggs 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

[Dietterich 1997] Dietterich, T. G. (1997). Machine-learning research. *AI magazine*, 18(4):97.

[Freund and Schapire 1995] Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer.

[Friedman 2002] Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

[Geurts et al. 2006] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

[Ghahramani 2004] Ghahramani, Z. (2004). Unsupervised learning. In *Advanced Lectures on Machine Learning*, pages 72–112. Springer.

[Jain and Dubes 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

[Leutenegger et al. 2011] Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE.

[Lloyd 1982] Lloyd, S. (1982). Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137.

[Lopes 2009] Lopes, A.; Avila, S. P. A. O. R. . A. A. (2009). A bag-of-features approach based on hue-sift descriptor for nude detection. In *Proceedings of the XVII European Signal Processing Conference (EUSIPCO)*, Glasgow, Scotland.

[Lowe 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Mikolajczyk and Schmid 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630.

[Rublee et al. 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE.

[Steel 2012] Steel, C. M. (2012). The mask-sift cascading classifier for pornography detection. In *Internet Security (WorldCIS), 2012 World Congress on*, pages 139–142. IEEE.

[Trzcinski et al. 2013] Trzcinski, T., Christoudias, M., Fua, P., and Lepetit, V. (2013). Boosting binary keypoint descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2874–2881. Ieee.

[Tuytelaars 2010] Tuytelaars, T. (2010). Dense interest points. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2281–2288. IEEE.

[Tuytelaars and Mikolajczyk 2008] Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280.

[Yang et al. 2007] Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of*

the international workshop on Workshop on multimedia information retrieval, pages 197–206. ACM.

[Yu and Han 2014] Yu, J.-J. and Han, S.-W. (2014). Skin detection for adult image identification. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 645–648. IEEE.

	1. Logar em um computador a. Para usar o NPDI *Find Porn*, você deve logar no computador que você deseja fazer a pesquisa.
	2. Plugar o *Pen Drive* NPDI *Find Porn* a. Plugar o *pen drive* com o sistema NPDI *Find Porn* em uma porta USB do computador.
	3. Abrir o arquivo NPDIFindPorn.bat a. Se o seu computador não pedir automaticamente para abrir o *pen drive* inserido no computador, no sistema operacional Windows, vá em "Meu Computador" e abra o *driver* manualmente. Depois dê um duplo clique em NPDIFindPorn.bat.
	4. Selecionar onde você deseja fazer a pesquisa e clicar em *"Start"* a. Você verá uma lista de drivers disponíveis para pesquisar por conteúdo pornográfico. Por padrão, o *driver* C: será selecionado. Você poderá escolher expandir qualquer *driver* e selecionar pastas específicas para pesquisar. b. Após selecionar um diretório de interesse, basta clicar no botão *"Start"* e aceitar a confirmação do diretório selecionado.
	5. Visualizar os resultados da pesquisa a. Você deve iniciar a consulta clicando no botão *"Start"* da janela que foi aberta, iniciando assim a etapa de busca por conteúdo pornográfico. Enquanto a análise é feita, é disponibilizada uma barra de progresso do processo. b. Após o fim da análise, o sistema mostra o tempo gasto para a análise da pasta selecionada e habilita o botão *"Open Report Folder"* para verificar os resultados da pesquisa. Nesse passo, o usuário deve confirmar que realmente gostaria de verificar o possível conteúdo pornográfico clicando no botão *"Yes"*. c. Após a confirmação de visualização do resultado da pesquisa, o sistema abrirá uma janela que possui uma lista com os nomes das imagens que foram classificadas como sendo pornográficas (canto esquerdo da janela aberta), assim como o caminho completo da imagem. O caminho da imagem dentro do computador é exibido no canto inferior da tela, após clicar no nome da imagem.

Figure 4. Funcionamento da Ferramenta NPDI *Find Porn*.

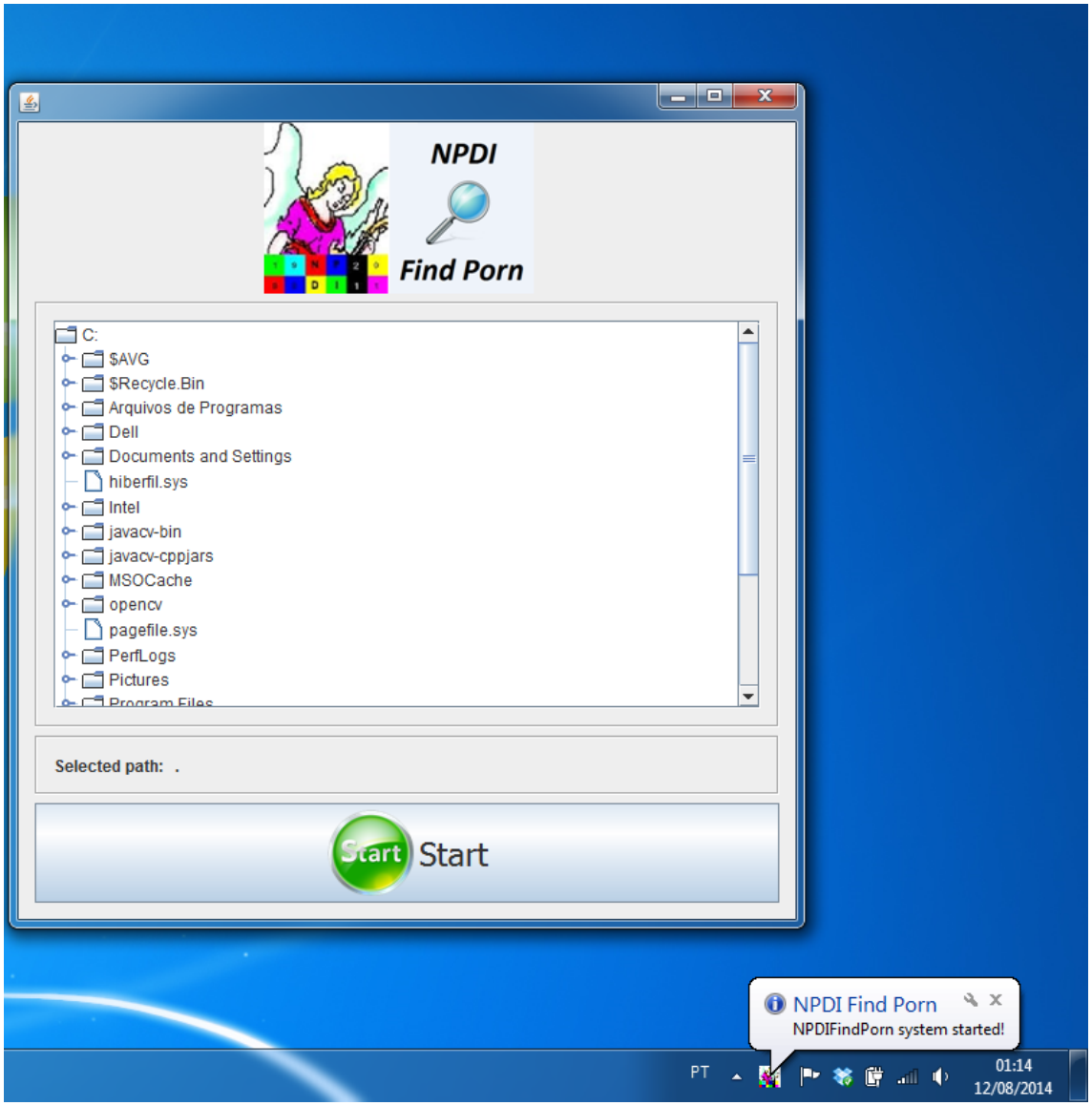


Figure 5. Tela inicial da ferramenta de detecção de conteúdo pornográfico NPDI *Find Porn*.

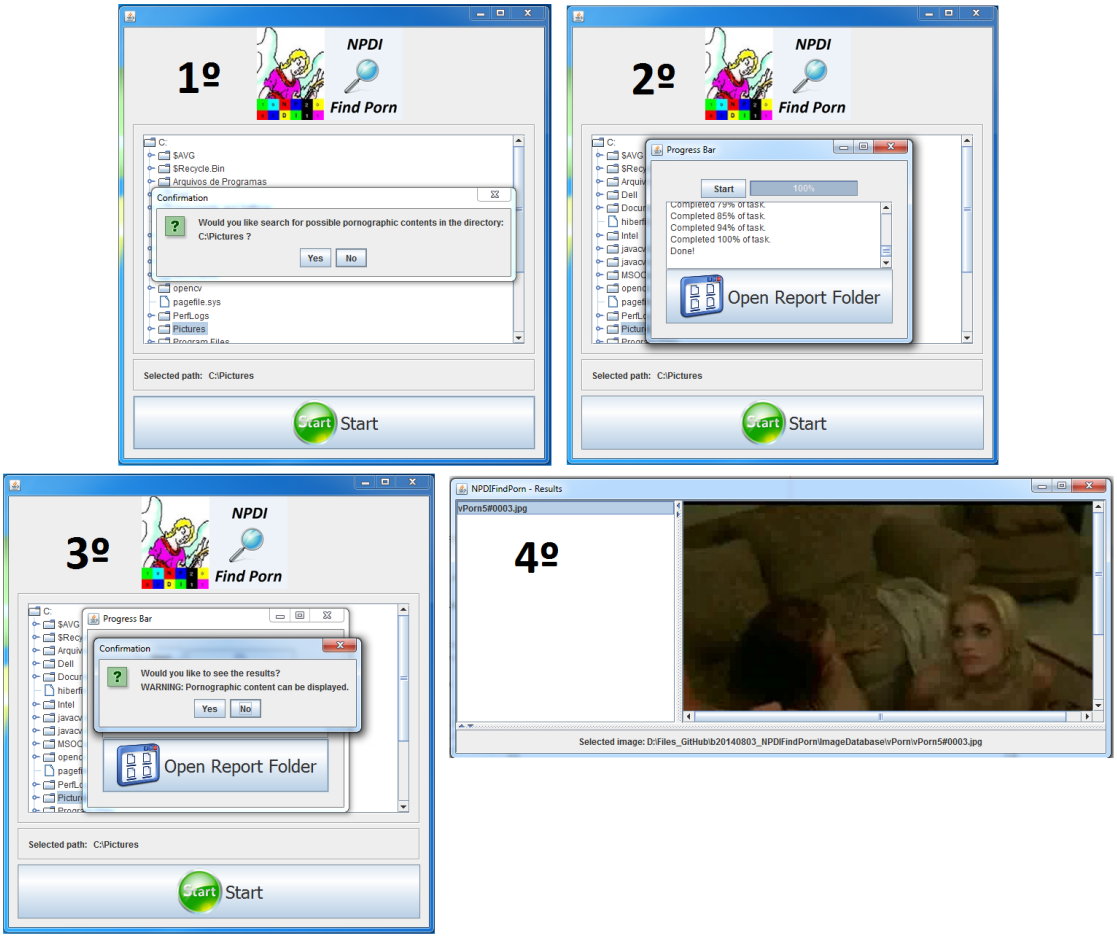


Figure 6. A ferramenta NPDI *Find Porn* em uso. Após selecionar um diretório a ser pesquisado, o sistema NPDI *Find Porn* busca por conteúdos pornográficos neste diretório e ao final da pesquisa exibe um relatório com um lista de imagens consideradas pornográficas, bem como o diretório no computador onde a imagem está armazenada (canto inferior da janela que exibe as imagens).

Linux Remote Evidence Colector – Uma ferramenta de coleta de dados utilizando a metodologia Live Forensics

Evandro Della Vecchia[12], Luciano Coral[3]

[1] Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre – RS – Brasil
evandro.pereira@pucrs.br

[2] Instituto Geral de Perícias (IGP) – Departamento de Criminalística
Porto Alegre – RS – Brasil
evandro-pereira@igp.rs.gov.br

[3] Universidade do Vale dos Sinos
São Leopoldo – RS - Brasil
lcoral.rs@gmail.com

Abstract. *This paper describes the stages of planning and implementation of a tool designed to collect volatile information in a process of forensic analysis. The text presents a view on the issue of electronic crimes, contextualize the practice of digital forensics, as well as its steps and methodologies, presenting a comparison between existing tools on the market and also describe in detail the tool developed as the result of the research, for the collection and analysis of data in an enterprise environment.*

Resumo. *Este artigo descreve as etapas do planejamento e implementação de uma ferramenta que se destina à coleta de informações voláteis em um processo de análise forense. O texto busca apresentar uma visão sobre a problemática dos crimes eletrônicos, contextualizar a prática da perícia digital na investigação de crimes eletrônicos, bem como suas etapas e metodologias aplicadas em uma perícia, apresentar um comparativo entre ferramentas já existentes no mercado e também descrever detalhadamente a ferramenta que foi desenvolvida para a coleta e análise de dados em um ambiente corporativo.*

1. Introdução

Diante do crescimento dos crimes praticados com a utilização de meios eletrônicos, também foi necessário o desenvolvimento de novas tecnologias de segurança da informação. Neste cenário, a área da Perícia Digital, que é a ciência que reconstitui as etapas de um crime eletrônico com o fim de proporcionar elementos de prova do delito, cresceu muito nos últimos anos.

Entretanto, apesar dos avanços, ainda lacunas, como a falta de ferramentas que possam ser utilizadas para análise de dados coletados de dispositivos com todos os sistemas operacionais existentes.

Após a pesquisa inicial para este trabalho foi constatado, entre outros problemas, a escassez de ferramentas para a realização de perícias em dados coletados em ambientes corporativos com sistema operacional Linux. Diante disto, considerando-se inclusive as deficiências dos instrumentos disponíveis, uma ferramenta que permite a coleta remota de evidências em computadores com sistema Linux foi desenvolvida.

Na Seção 2 é apresentada a problemática dos crimes cibernéticos, sua definição e os delitos mais frequentemente cometidos por este meio. Na Seção seguinte são apresentados conceitos da Perícia Digital, incluindo as metodologias *Live Forensics* e *Post Mortem Forensics*. Na Seção 4 foi realizado um comparativo entre duas ferramentas existentes para o auxílio da perícia digital: o EnCase e o Helix. Depois de realizado o comparativo entre tais ferramentas, critérios foram estabelecidos para o desenvolvimento de uma nova (a LREC), para atender outras necessidades na realização de uma análise forense (Seção 5).

2. Crimes Cibernéticos

Segundo Della Vecchia (2014), o crime cibernético, ou cyber crime, é definido como um ato ilegal que utiliza o computador e a acessibilidade de informações, principalmente através da Internet para planejar e/ou realizar suas atividades. Tais atividades englobam: a exploração de computadores para acesso a informações confidenciais, fraudes eletrônicas, redes de pedofilia, espionagem, roubo de identidade, entre outros.

Existem algumas classificações, porém a que tem encontrado maior ressonância entre os doutrinadores do Direito é a que divide os crimes cibernéticos em próprios (exclusivamente cibernéticos) e impróprios (abertos), conforme descrito abaixo.

- Crimes Cibernéticos Próprios ou Exclusivamente Cibernéticos: exigem e dependem necessariamente da utilização de ambiente computacional. A execução depende da utilização de recursos tecnológicos como meio e objeto da prática delituosa, ou seja, sem a utilização de tais recursos não seria possível cometer o crime. Alguns exemplos são: a criação e disseminação de códigos maliciosos, ataques de negação de serviço e invasão/destruição de banco de dados. No Brasil, a Lei 12.737/2012, conhecida como "Lei Carolina Dieckmann", trata de alguns crimes cibernéticos próprios;

- Crimes Cibernéticos Impróprios ou Abertos: o ambiente computacional é utilizado como meio para execução da conduta ilícita, porém o resultado poderia ser obtido de forma comum, sem o uso da tecnologia. Alguns exemplos são os crimes contra a honra, ameaça, falsificação, estelionato, furto, entre outros.

Segundo dados publicados pelo Cert.br, no ano de 2013 foram notificados 352.925 incidentes de segurança. Nos anos anteriores a quantidade foi maior, conforme mostra a Figura 01. Os incidentes mais comuns reportados são varreduras (*scan*) e fraudes, tais como roubo de dados de cartões de crédito.

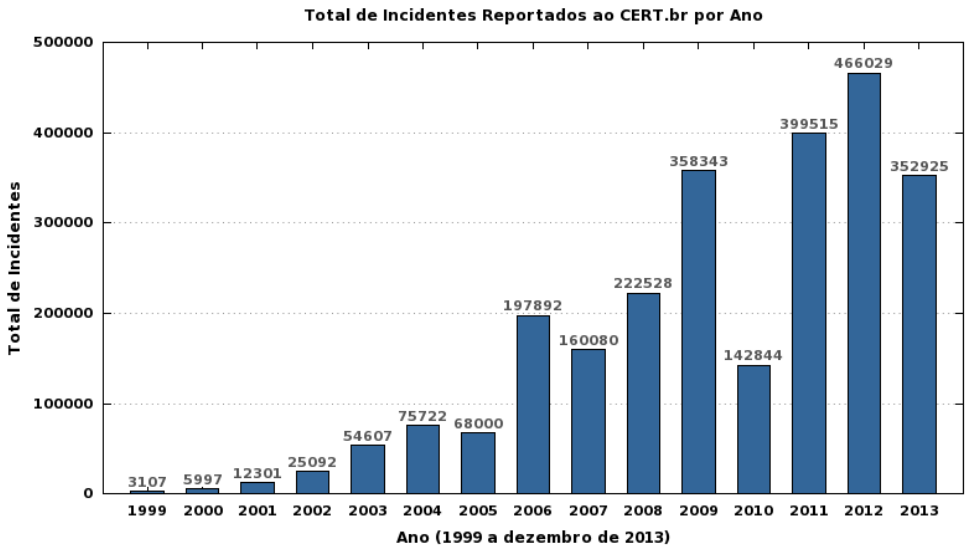


Figura 1. Estatística anual Cert.br. Fonte: <http://www.cert.br/stats/incidentes/>.

3. Perícia Digital

A Perícia Digital, também conhecida como Forense Computacional, Forense Digital, entre outras expressões, é a área da Perícia destinada a auxiliar na elucidação de crimes praticados por meios eletrônicos.

Por meio da utilização de métodos científicos, o perito digital é capaz de identificar, preservar, analisar e documentar evidências encontradas em computadores e dispositivos eletrônicos, por meio dos quais pode ter sido praticado algum delito.

Há basicamente duas metodologias: a *post mortem forensics* e a *live forensics*, sendo a primeira utilizada quando os equipamentos já estão desligados e a segunda quando há um flagrante com equipamentos ligados e é possível então coletar dados da memória volátil também.

A Figura 2 mostra cinco etapas possíveis para a aplicação da perícia digital, seguindo boas práticas. Após uma breve explicação de cada uma será apresentada.

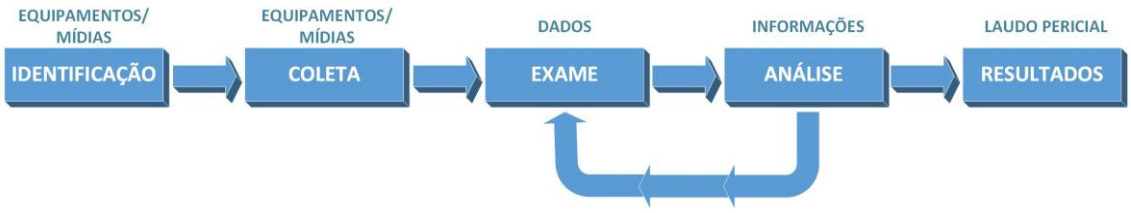


Figura 2. Etapas da Perícia Digital. Fonte: Della Vecchia (2014).

Na primeira etapa, os equipamentos e mídias são identificados, fotografados, etiquetados e lacrados, caso a metodologia seja a *post mortem*, ou apenas identificados caso seja a *live forensics*. Na fase de coleta as mídias são protegidas contra escrita (*post mortem*), os dados são copiados bit a bit (cópia integral) e o *hash* é gerado para a garantia de integridade. Na fase de exame, são aplicados filtros, buscas por palavras-

chave e outros procedimentos na cópia integral. Na etapa seguinte as ocorrências obtidas como resultado da fase de exame são analisadas pelo perito, que por fim relata o Laudo Pericial na etapa Resultados.

4. Ferramentas Relacionadas

Nesta Seção serão apresentadas duas ferramentas que são utilizadas em perícias digitais. Tais ferramentas foram analisadas no intuito de embasar o desenvolvimento de uma nova ferramenta voltada à prática da perícia digital, capaz de realizar a coleta de dados voláteis em sistemas Linux.

Neste estudo, foi analisada a ferramenta comercial EnCase[1] e a distribuição Linux Helix[2], ambas completas e capazes de atender aos requisitos de todas as etapas de uma perícia digital.

As ferramentas analisadas possuem características diferentes, que as tornam mais ou menos eficazes. Sendo assim, foram definidos alguns critérios para compor um padrão de desenvolvimento para a nova ferramenta. Foram observados os seguintes critérios:

- metodologia de trabalho: *Live Forensics*, *Post Mortem Forensics*, ou ambas;
- sistema operacional alvo: Windows, Linux ou ambos;
- arquitetura: modular ou não-modular;
- forma de realização da coleta: local ou remota;
- garantia de integridade: utilização de algoritmo de hash;
- apresentação do resultado: classificada como apresentação em vídeo ou relatório de evidências.

4.1 EnCase

Desenvolvido pela Guidance Software, o EnCase é uma ferramenta gráfica muito poderosa para investigar crimes digitais. Atualmente a ferramenta está na versão 7, que permite entre outras funcionalidades, a possibilidade de realizar toda a operação de forma remota. O funcionamento básico do EnCase inclui a criação das cópias (imagem fiel da mídia questionada), validação da mídia (aplicação de algoritmos de *hash*), análise do conteúdo (avaliação das evidências coletadas) e apresentação dos resultados (geração de relatórios).

De acordo com os critérios estabelecidos, o EnCase foi classificado conforme mostra a Tabela 01:

Tabela 01. Análise do Encase.

Metodologia de trabalho	Post Mortem
Arquitetura	Modular

[1] Disponível em <https://www.guidancesoftware.com/>.

[2] Disponível em <http://www.e-fense.com/products.php>.

Sistema alvo	Windows / Linux / Solaris
Forma de coleta	Local / Remota
Garantia de integridade	Algoritmo MD5

4.2 Helix

O Helix é uma distribuição Linux que pode ser baixada e distribuída livremente sob a forma de um live CD (até a versão 2009). Esta distribuição possui uma característica muito interessante, pois opera em modos de trabalho diferentes dependendo de como foi iniciada em um equipamento. O Helix pode ser acionado durante o processo de inicialização de um equipamento ou ser carregado a qualquer momento, em uma máquina em funcionamento, acionando-se a unidade de CD/DVD.

Quando o live CD é acionado durante a inicialização de um computador, o Helix assume uma interface baseada na distribuição Knoppix do Linux. Por meio desta interface, é possível acessar diversas ferramentas para forense que estão agregadas ao Helix, tais como o Autopsy e o PyFlag.

O outro modo de trabalho suportado pelo Helix é acionado quando o live CD da ferramenta é inserido em um equipamento em funcionamento que esteja usando o sistema operacional Windows.

De acordo com os critérios estabelecidos, o Helix foi classificado conforme mostra a Tabela 2.

Tabela 2. Análise do Helix.

Metodologia de trabalho	Post Mortem e Live
Arquitetura	Não-modular
Sistema alvo	Windows / Linux
Forma de coleta	Local
Garantia de integridade	Algoritmo MD5

5. Linux Remote Evidence Colector (LREC)

A ferramenta LREC tem como principal objetivo atender as necessidades de um perito no que tange a metodologia *Live Forensics* em um ambiente corporativo que utilize o sistema operacional Linux. Tendo como premissa a preservação da integridade dos dados, o uso desta ferramenta procura reduzir a probabilidade de erros operacionais, por parte do perito, que venham a comprometer o andamento normal da perícia.

Foi desenvolvida em shell script e utiliza recursos de gerenciamento gráfico providos pelo Zenity[3]. A interface principal do LREC pode ser visualizada na Figura 3.

[3] Ferramenta que recebe os parâmetros do código desenvolvido em *shell script*, tornando possível a utilização de janelas, caixas de diálogo, calendários e uma série de outros recursos que permitem oferecer ao usuário uma interface operacional mais amigável.

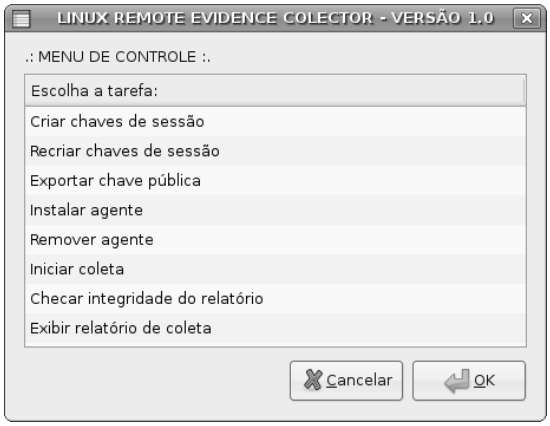


Figura 3 - Interface gráfica do LREC.

5.1 Arquitetura do LREC

Visando atender à demanda exigida pelo processo de análise forense em ambientes corporativos que contam com um número elevado de servidores, o LREC tem uma arquitetura baseada em dois componentes: (a) gerente; e (b) agente de coleta.

A Figura 4 apresenta uma visão simplificada da arquitetura do LREC. Nela é possível observar o equipamento do perito interagindo com os equipamentos questionados por meio de uma conexão segura.

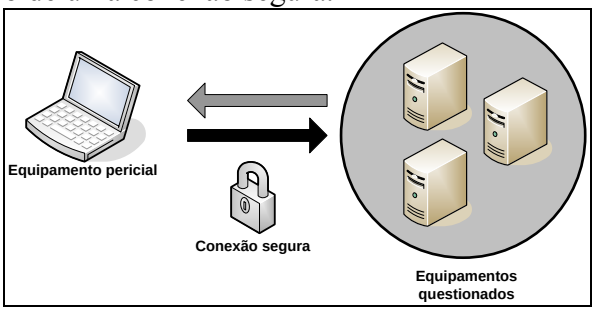


Figura 4 - Visão simplificada do LREC.

O agente é o responsável pela coleta de informações nos equipamentos questionados. Este componente é totalmente desprovido de interface, uma vez que o mesmo opera em modo *background* no sistema operacional. A arquitetura permite que o agente seja instalado em vários equipamentos, para permitir uma coleta simultânea.

Com o intuito de evitar qualquer comprometimento do sistema causado pela existência de *rootkits*, o LREC conta com um conjunto próprio de ferramentas, sendo assim, em nenhum momento o LREC fará chamada a alguma ferramenta nativa do sistema analisado.

O gerenciamento de todas as funções do LREC é realizado pelo gerente, que é instalado no equipamento do perito e oferece uma interface simplificada que controla as funções do sistema. Este componente contém dois arquivos: o lrec.sh e o connect.sh. O arquivo lrec.sh tem o código que controla a interface principal do LREC, realizando diversas chamadas de função que são mantidas em uma biblioteca externa. O connect.sh

é uma biblioteca de funções que são consultadas pelo lrec.sh durante a execução da ferramenta.

As funções contidas na biblioteca permitem o gerenciamento completo das operações do LREC. Por meio dessas funções é possível: (a) gerenciar a troca das chaves de sessão entre os servidores; (b) instalar e desinstalar o agente de coleta; (c) iniciar a coleta de evidências; (d) apresentar o relatório de coleta; entre outras funções.

5.2 Funcionamento do LREC

Para um melhor entendimento do modo de operação do LREC pode-se observar a Figura 5, que descreve o funcionamento do LREC em quatro etapas, identificadas numericamente, da seguinte forma: (1) o equipamento do perito compartilha a sua chave pública com o equipamento questionado; (2) é estabelecida uma conexão segura entre os equipamentos e é iniciada a instalação do módulo de agente de coleta no equipamento questionado; (3) após a conclusão da instalação do módulo agente é iniciada a coleta das evidências no equipamento questionado; e (4)o equipamento do perito recebe o relatório de coleta enviado pelo módulo agente.

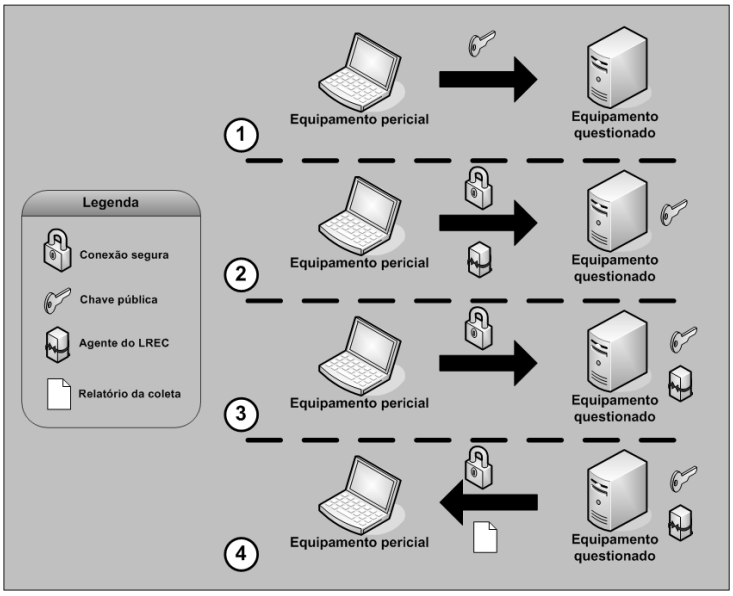


Figura 5 – Etapas do funcionamento do LREC.

5.3 Funções de gerenciamento do LREC

Como o LREC é uma ferramenta de coleta remota de evidências, os dados transmitidos entre o equipamento questionado e o equipamento do perito são cifrados, visando para garantir a confidencialidade. Para tanto, é necessário criar o par de chaves de sessão na máquina do perito e exportar sua chave pública para o equipamento questionado, procedimento explicado a seguir.

A função de criação de chaves do LREC cria o par de chaves utilizando o algoritmo RSA[4]. A figura 6 mostra o processo de criação do par de chaves.

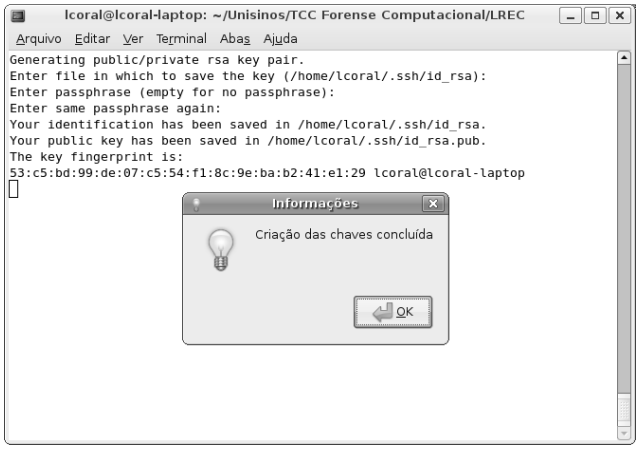


Figura 6 - Criação do par de chaves RSA.

O resultado desta operação pode ser visto na figura 7. Os arquivos gerados são o "id_rsa" (chave privada) e o "id_rsa.pub" (chave pública). Para que o LREC estabeleça a conexão, a chave pública do equipamento do perito deve ser exportada para os equipamentos remotos. Caso haja o comprometimento da chave privada, há a possibilidade de recriação do par de chaves.

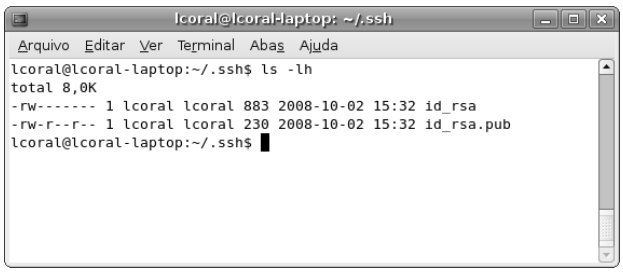


Figura 7 – O par de chaves gerado.

Durante o processo de exportação da chave pública, o perito deverá informar o endereço IP do equipamento para o qual deseja publicar a chave, conforme mostra a figura 8.

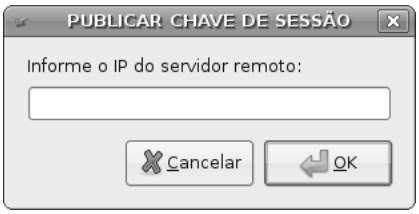


Figura 8 - Digitação do endereço IP do equipamento remoto.

[4] RSA é um algoritmo de encriptação de dados, que deve o seu nome a três professores do Instituto MIT, Ron Rivest, Adi Shamir e Len Adleman.

Após ter realizado as etapas de criação de chaves e exportação da chave pública, o componente de gerenciamento do LREC consegue estabelecer conexão com os equipamentos remotos a serem analisados. Até então o LREC restringe-se apenas a conectar-se com estes equipamentos, não sendo capaz de realizar nenhuma coleta de evidências.

Para tornar possível a coleta de evidências nos equipamentos remotos que já possuem a chave pública do perito, é necessário instalar o agente de coleta. Esta instalação é realizada pelo próprio componente de gerenciamento do LREC, bastando apenas informar o endereço IP do equipamento de destino, conforme mostra a Figura 9.

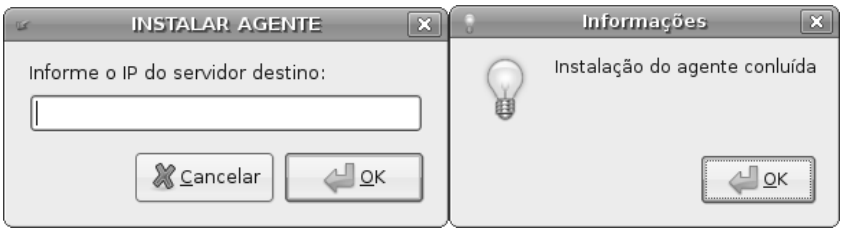


Figura 9 - Processo de instalação do agente LREC.

O processo de instalação cria no equipamento remoto uma estrutura que contém o agente de coleta que será usado na perícia, bem como o conjunto de ferramentas próprias usadas pelo mesmo. Esta medida se faz necessária para garantir que as ferramentas usadas estão livres de interferências provocadas por *rootkits*. A figura 10 mostra a estrutura criada pelo LREC no equipamento remoto, após a instalação do agente de coleta. É possível observar o agente de coleta representado pelo arquivo coleta.sh e pelo diretório tools, que contém as ferramentas usadas pelo agente.

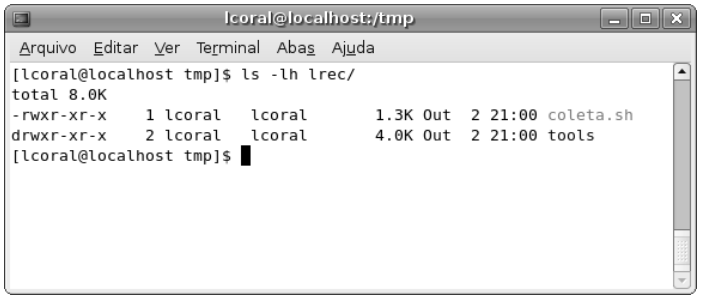


Figura 10 - Estrutura criada após a instalação do agente.

O início da coleta é comandado de forma manual ou de forma agendada. Para o agendamento, é utilizado o serviço *cron* (arquivo de configuração *crontab*). Desta forma, é possível realizar a coleta em vários períodos em um mesmo dia.

Ao término do processo de coleta, seja ele executado pela forma manual ou agendada, um relatório de coleta será criado e o mesmo enviado para o equipamento do perito. Todos os relatórios seguem o padrão de nomenclatura "computador.txt" e são armazenados no subdiretório "coletados", onde está instalado o LREC (Figura 11).

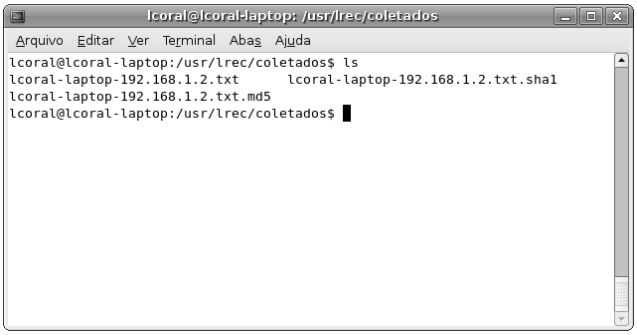


Figura 11 – Relatório gerado pelo LREC.

Para garantia de integridade de um relatório armazenado no equipamento do perito existe a opção de verificação através da tela mostrada na Figura 12.

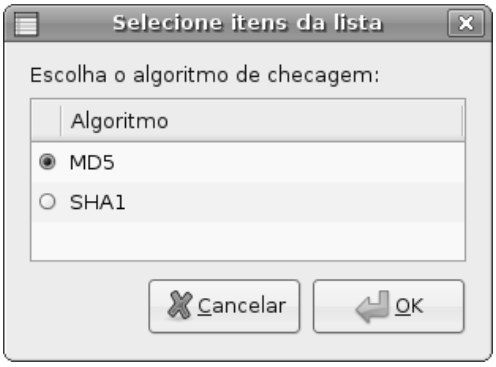


Figura 12 - Escolha do algoritmo de *hash*.

A última função do componente de gerenciamento do LREC diz respeito à apresentação dos resultados coletados sob a forma de relatório, conforme mostra a Figura 13.

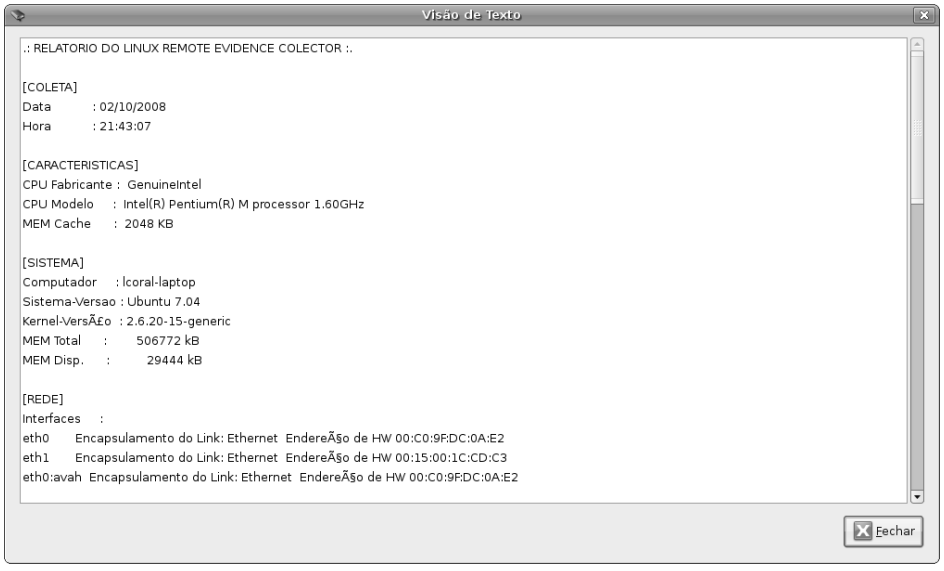


Figura 13 – Visão parcial do relatório de coleta.

5.4 Ambiente de Testes do LREC

Como forma de validar as funcionalidades do LREC, bem como sua real contribuição para a área da Perícia Digital, foi criado um cenário para a realização dos testes necessários. O cenário adotado é composto por duas máquinas, sendo uma delas física e a outra virtual. Em ambas as máquinas foram utilizadas distribuições Linux, mas com diferentes versões de kernel.

Na máquina física, foi instalada a distribuição RedHat, versão 9, utilizando o kernel 2.4 do Linux. Já na máquina virtual, foi instalada a distribuição Ubuntu, versão 8.04, utilizando o kernel 2.6 do Linux. Trata-se de versões antigas, justamente para garantir melhor funcionalidade, mesmo com versões não atuais.

Em ambos os sistemas, o LREC realizou suas atividades conforme previsto em sua especificação, embora tenha sido necessário atualizar a versão da biblioteca libc no sistema com a versão 2.4 do kernel do Linux.

6. Considerações Finais

A elaboração deste trabalho foi baseada em pesquisa e implementação. A pesquisa teve como foco a prática de crimes cibernéticos e o emprego de técnicas de perícia digital para a resolução de muitos desses crimes.

Como forma de reduzir o risco de falhas humanas durante a etapa inicial de uma perícia (coleta de dados), foi proposta e desenvolvida uma ferramenta que se destina a coletar informações voláteis, em sistemas Linux.

Diante do resultado dos testes realizados em ambiente apropriado para este fim, a LREC atendeu às especificações do projeto, ou seja, foi capaz de coletar remotamente evidências de equipamentos para posterior análise.

Um dos maiores problemas enfrentados durante a elaboração da ferramenta foi mensurar a interferência que o LREC causa nos equipamentos periciados, já que um agente de coleta necessita ser instalado neles. Durante os testes, os percentuais de uso de processamento e consumo de memória foram medidos e apresentaram alterações insignificantes durante a execução do agente de coleta. Quanto às alterações de disco, foi apontado que a instalação do agente de coleta ocupa cerca de 490 kilobytes e que o relatório de coleta ocupa 40 kilobytes do disco no computador periciado.

Foi analisado um método para realizar um *dump* de memória da máquina periciada, mas a implementação deste acabou não sendo realizada, porque causaria muitas alterações no ambiente periciado.

Como o sistema periciado é Linux, cogitou-se a possibilidade da realização deste *dump* por meio do uso do comando *dd*, que realiza a cópia bit-a-bit dos dados. Ocorre que, em função do grande volume de dados que seria gravado em disco, esta alternativa foi descartada já que realmente iria modificar muito o cenário inicial da perícia.

Outra alternativa seria a realização da cópia do arquivo que armazena o conteúdo da memória em sistemas Linux, o *kcore*. Esta prática também se mostrou inviável já que a cópia de um arquivo com tamanho na ordem de gigabytes reduz drasticamente o desempenho da rede, o que aumenta muito o tempo necessário para a realização da mesma. Mesmo com a ausência desta funcionalidade o LREC representa uma

contribuição para a área da Perícia Digital por ser uma ferramenta simples, capaz de operar remotamente e auxiliar o perito em coletas *Live*.

Referências Bibliográficas

Brezinski, D.; Killalea, T. (2002) "Guidelines for Evidence Collection and Archiving", Request for Comments: 3227. <Disponível em https://www.ietf.org/rfc/rfc3227.txt>. Acesso em Set. 2014.

Cert (2014) "Estatísticas dos Incidentes Reportados ao CERT.br", Disponível em <http://www.cert.br/stats/incidentes/>. Acesso em Set. 2014.

Della Vecchia, E. (2014) "Perícia Digital - Da investigação à análise forense", Campinas: Millennium. 296p.

Della Vecchia, E.; Weber, D.; Zorzo, A. (2013) "Anti-Forense Digital: conceitos, técnicas, ferramentas e estudos de caso", Minicursos / XIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Porto Alegre: Sociedade Brasileira de Computação.

Guidance (2014) "Encase Forensic v7", Disponível em <https://www.guidancesoftware.com/products/Pages/encase-forensic/overview.aspx?cmpid=nav>. Acesso em Set. 2014.

Helix (2014) "E-fense Carpe Datum", Disponível em <http://www.e-fense.com/products.php>. Acesso em Set. 2014.

Hoelz, B. W. P.; Mesquita, F. I.; Auler, P.(2011) "Live forensics em ambiente Microsoft Windows", Minicursos / XI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Porto Alegre: Sociedade Brasileira de Computação.

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

WTE – I Workshop Tecnologia Eleitoral Computacional

Critérios para Avaliação de Sistemas Eleitorais Digitais

Amilcar Brunazo Filho[1], Mario A. Gazziro[2]

[1]CMIND – Comitê Multidisciplinar Independente

[2]UFABC – Universidade Federal do ABC

amilcar@brunazo.eng.br, mario.gazziro@ufabc.edu.br

Abstract. *Since the first applications with electronic voting machines in the 90s models and standards have been developed but little is discussed what should be the criteria for classification and comparison of such models. This paper proposes a set of basic concepts necessary and their comparison criteria that can be applied in the design and evaluation of digital electoral systems. A table of compliance is also presented for some models in use in several countries in Latin America.*

Resumo. *Desde as primeiras aplicações com máquinas eletrônicas de votação nos anos 90, alguns modelos e padrões já foram desenvolvidos, mas é pouco discutido quais deveriam ser os critérios de classificação e de comparação de tais modelos. Neste trabalho é proposto um conjunto de conceitos básicos necessários e de critérios de comparação que possam ser aplicados no projeto e na avaliação de sistemas eleitorais digitais. Também é apresentada uma tabela de conformidade para alguns modelos em uso em diversos países da América Latina.*

1. Introdução

Já em 1993, quando se iniciava o uso de equipamentos eletrônicos de votação em eleições oficiais de alguns países, Peter G. Neumann apresentou uma proposta de critérios de segurança para ser aplicada na avaliação de sistemas de eleição eletrônica [Neumann 1993] que envolvia uma longa lista de itens sobre confidencialidade, integridade, disponibilidade, confiabilidade, segurança e muitos outros.

Desde então, foram desenvolvidos novos recursos de hardware e de software e novas infraestruturas de comunicação tiveram grande crescimento. Simultaneamente, vários países passaram a testar e adotar sistemas eleitorais digitais isolados ou conectados pela Internet. Esse crescimento de recursos e de sistemas ocorreu de forma independente e sem coordenação, gerando uma grande variedade de modelos com características bem diferenciadas quanto ao atendimento dos critérios de segurança e de confiabilidade.

Na área do voto digital, critérios de confiabilidade que parecem essenciais para uns, são simplesmente ignorados por outros e os esforços de padronização ainda são modestos e insipientes, tendo se encontrado obstáculos e resistências inesperados, que acabam retardando a produção de normas locais e internacionais.

O presente trabalho apresenta um pequeno conjunto de conceitos fundamentais e critérios para classificação e avaliação de confiabilidade de sistemas eleitorais digitais e foi desenvolvido a partir da experiência dos autores no estudo e acompanhamento dos sistemas usados em países da América Latina, com destaque dos sistemas usados no Brasil, na Venezuela e na Argentina.

Em nenhum desses três casos se usa o voto pela Internet porque a legislação dos três países, em respeito ao Princípio da Inviolabilidade do Voto, exige que o ato de votar ocorra em um local isolado e protegido contra acesso de terceiros, nas chamadas cabines indevassáveis. Apenas a transmissão dos resultados parciais, resultantes das contagens dos votos digitais dados em máquinas isoladas, pode ocorrer pela Internet. Assim, a descrição de modelos e as tabelas de conformidade apresentadas no final do trabalho consideram apenas os modelos de equipamentos de gravação do voto sem conexão com a Internet durante o ato de votação.

Ainda com relação à votação pela Internet, há fortes restrições nos meios acadêmicos quanto à sua confiabilidade técnica. Em 2008, por solicitação da ONG americana *Verified Voting*, uma comissão de mais de trinta especialistas em TI elaborou um estudo [CT 2012] contrário aos modelos de votação pela Internet existentes por causa dos enormes desafios que ainda não foram tecnologicamente resolvidos para justificar a confiabilidade técnica de tais sistemas.

Por estes motivos, o presente estudo não abrange o que genericamente é chamado de "sistemas eleitorais pela Internet".

Experiências e observações com voto digital em eleições oficiais sempre estarão inseridas em ambientes e situações influenciadas pelas peculiaridades culturais e políticas locais e até mesmo pelas fortes cargas emocionais que eleições oficiais acendem. Dessa forma, o presente trabalho destacará conceitos fundamentais e critérios básicos que os autores entendem ser necessários ser aplicados nas realidades sociais e políticas de onde derivam suas observações.

2. Conceitos Fundamentais e Critérios Básicos para o Voto Digital

2.1. Confiabilidade de Sistemas Eleitorais

O conceito de confiabilidade de sistemas eleitorais envolve duas faces distintas de igual importância:

Confiança técnica: determinada por avaliações e medidas objetivas;

Confiança subjetiva: baseada no sentimento pessoal.

Em um processo eleitoral, pode-se tentar conquistar a confiança subjetiva dos atores, transferindo a confiança que depositam nas autoridades eleitorais para a confiança no processo que as autoridades administram.

Mas, em processos eleitorais eletrônicos, onde pequenas falhas podem passar despercebidas provocando grandes desvios no resultado, e ainda, onde os interesses das partes envolvidas são crescentes e podem ser divergentes, cada vez mais se torna necessário conquistar a confiança subjetiva dos eleitores e dos candidatos por meio da demonstração objetiva da confiabilidade técnica do processo.

Em 2010, Wolter Pieters [Pieters 2010] propôs que as palavras inglesas "*confidence*" e "*trust*", quando usadas em referências a processos eleitorais, tivessem significados similares aos de confiança subjetiva e de confiança técnica, respectivamente, e apresentou o quadro a seguir, atribuído a Niklas Luhmann:

Tabela 1. Modelos de auto-confiança subjetiva e técnica proposto por Luhmann

	"Confidence" **(Confiança subjetiva)**	**"Trust"** **(Confiança técnica)**
Tipo de confiança	Inconsciente	Consciente
Interpretação	Alternativas não percebidas	Comparação de alternativas
Ação	Sem decisão	Decisão / Escolha
O que os cientistas desejam	Minimizar	Maximizar

A proposta é que, em processos eleitorais eletrônicos, deve-se procurar minimizar a necessidade de confiança subjetiva dos eleitores e maximizar a confiança tecnicamente demonstrada.

Também é comum o entendimento que a melhor via para avaliar e demonstrar a confiabilidade técnica em um processamento eletrônico de votos é a TRANSPARÊNCIA de todos os atos de votação, de registro dos votos (gravação digital) e de contagem dos votos digitais.

2.2. Soberania dos Eleitores e dos Candidatos

Em eleições, os direitos soberanos e prioritários pertencem aos eleitores (direito de votar) e aos candidatos (direito de ser votado). Tais direitos devem prevalecer e não podem ser restringidos pelos direitos dos demais participantes do processo eleitoral, como os administradores, juízes, auditores, fornecedores, programadores e demais operadores, os quais têm por DEVER garantir os direitos eleitorais do cidadão.

Para exercer seu direito de forma soberana, ELEITOR e CANDIDATO devem poder FISCALIZAR COM RECURSOS PROPRIOS e EFETIVOS todos os procedimentos eleitorais, como votação, registro de voto, apuração, etc., como bem explanado pelo Procurador da República Celso Antônio Três [Três 2002], durante o Seminário do Voto Eletrônico promovido pela Câmara dos Deputados em maio de 2002, quando expôs:

"Contudo, mesmo fosse cientificamente possível garantir a segurança técnica, isso não seria suficiente. Impõe-se disponibilizar ao cidadão, através de suas faculdades normais, motu próprio, a possibilidade de sindicar a devida observância à sua vontade eleitoral.

A Constituição da República, de forma lapidar e definitiva, estabelece a pedra fundamental do Estado Brasileiro, após certificar que "... todo o poder emana do povo..." (art. 1º,§ único, da C.F.), diz que "a soberania popular é exercida pelo sufrágio universal e pelo voto direto e secreto ..."(art. 14, "caput", da C.F.).

De sua parte, um dos sustentáculos do Direito Constitucional, vital a conferir efetividade aos preceitos fundamentais, é a conhecida teoria/doutrina dos poderes implícitos, traduzida pelo extraordinário Mestre Paulo Bonavides, ao dizer que "... na interpretação de um poder, todos os meios ordinários e apropriados a executá-lo são considerados sempre parte do próprio poder..."(Curso de Direito Constitucional, Malheiros, 10ª edição, p. 432).

De que vale um poder, uma prerrogativa, desprovido dos instrumentos necessários à sua efetivação?!?!?

Soberania pressupõe poder supremo. Onde está a supremacia do povo em um processo cuja apuração não é instrumentado por mecanismos que permitam-lhe certificar-se da soberania de sua vontade?!?!?. Pior. Sequer os agentes operadores, Membros da Justiça Eleitoral, do Ministério Público, dos Partidos Políticos, Candidatos, são, diretamente, dele dotados. Apenas assistidos por técnicos.

Soberano que não é instrumentado a fiscalizar o exercício de sua soberania não é soberano.

É inerente, "ratio essendi" da soberania popular, que todo o processo eleitoral, alistamento, registro de candidaturas, propaganda política, votação, apuração, diplomação, etc., sejam aferíveis pelo titular dessa soberania, o povo. Aferíveis, diga-se, por todo o eleitorado, desde o mais rutilante PhD até o excluído analfabeto"

O mesmo conceito foi acatado pelo Tribunal Constitucional da Alemanha, ao julgar um processo em 2009, onde se analisava a constitucionalidade de sistemas de voto puramente digital. Num longo acórdão [TCFA 2009], a corte suprema alemã criou jurisprudência, demarcando princípios e fundamentos sobre o uso de máquinas de votar, dos quais se destaca o seguinte, conforme tradução para o português realizada pelo CMind [CMind 2010]:

"Princípios

2. Na utilização de máquinas eletrônicas de votar, é necessário que o cidadão, que não possui experiência especial sobre o assunto, possa controlar de forma confiável os passos essenciais da ação de votar e da aferição dos resultados.

Fundamento 156

As principais etapas no processamento dos dados pelas máquinas de votar não poderiam ser entendidas pelo público. Como a apuração é processada apenas dentro das máquinas, nem os oficiais eleitorais, nem os cidadãos interessados no resultado podiam conferir se os votos dados foram contados para o candidato correto ou se os totais atribuídos a cada candidato eram válidos. Com base num resumo impresso ou num painel eletrônico, não era suficiente conferir o resultado da apuração dos votos na central eleitoral. Assim, foi excluída qualquer conferência pública da apuração que os próprios cidadãos pudessem compreender e confiar sem precisar de conhecimento técnico especializado."

Como consequência direta da extensão desse conceito, deve ser destacado que a participação de técnicos e da academia na validação de um processo eleitoral digital, embora seja prática desejável, não pode ser imposta como substituta válida ao direito do

eleitor comum compreender e auditar o registro e o destino do seu voto usando seus próprios conhecimentos e recursos.

A experiência com o processo eleitoral brasileiro serve como exemplo da necessidade de prevalência do direito do eleitor fiscalizar *motu-próprio,* sobre possíveis auditorias desenvolvidas por especialistas e acadêmicos, como o caso das fragilidades no software das urnas eletrônicas encontradas em 2012 pela equipe do professor Dr. Diego Aranha [Aranha 2012] depois do mesmo software das urnas eletrônicas ter passado por análise e escrutínios por várias equipes de auditores e especialistas contratados pela autoridade eleitoral desde 2002.

E tais tipos de fragilidades ainda persistem, como um dos autores do presente trabalho pôde constatar *in-loco* na Cerimônia de Apresentação dos Sistemas, no TSE em 2014.

A tentativa de substituir uma forma de auditoria contábil simples, feita pelos eleitores e candidatos, por uma validação e certificação exaustiva do software usado em mais de 400 mil equipamentos no dia da eleição, é complexa e financeiramente proibitiva para os agentes autorizados (Ministério Público, OAB e partidos políticos), como mais uma vez ficou comprovado nessa última Cerimônia de Apresentação dos Sistemas, onde tais agentes tradicionalmente não participam por alegada falta de recursos.

O Princípio da Soberania dos Eleitores e Candidatos tem forte influência na avaliação de sistemas eleitorais existentes, pois é muito frequente que auditorias feitas por especialistas sejam impostas como substitutas da possibilidade de auditoria pelo eleitor comum e pelos candidatos, sendo apresentadas como importante marco de confiabilidade quando, na realidade, estão desatendendo um direito fundamental de cidadania.

2.3. Tripartição de Poderes

O conceito de tripartição dos poderes, que surgiu na antiga Grécia e agora é adotado em todas as repúblicas modernas (pós Revolução Francesa), foi formalizado por Montesquieu no final do Sec. XVIII, e divide as funções de governo em três poderes independentes, harmônicos e autorreguladores: Legislativo, Executivo e Judiciário

Seu objetivo é evitar o abuso de poder, o autoritarismo, o absolutismo e o corporativismo, dando transparência aos atos de governo.

No Brasil, a tripartição dos poderes é estabelecida pelo Art. 2º da Constituição Federal de 1988, mas não vigora no processo eleitoral, onde uma mesma entidade, apesar de chamada de Justiça Eleitoral, detém as funções normativas, administrativas, fiscalizatórias e judiciais, decorrentes do art. 1º e Parag. Único do Código Eleitoral (Lei 4.737/65).

No 1º Relatório CMind [CMind 2010] é analisado e apresentado detalhes de como não há controle jurisdicional externo sobre as autoridades eleitorais brasileiras, resultando em abuso de poder, falta de transparência administrativa e quebra da imparcialidade judicial que, no seu exercício, acabam por restringir diretamente a efetividade dos direitos do eleitor e do candidato, acima citados.

2.4. Princípio de Inviolabilidade Absoluta do Voto

Há mais de cem anos foram criados os conceitos de inviolabilidade do voto e o de votação em cabines indevassáveis, para impedir a coação de eleitores, pois esta é uma modalidade de fraude eleitoral muito insidiosa, com grande potencial de modificação da verdade eleitoral. Assim, o sistema eleitoral deve impedir que terceiros possam identificar o autor de um voto. Também, ao eleitor não deve ser possível provar, para terceiros, em quem votou.

Entre os procedimentos criados para defender esse princípio, se insere o conceito de cabines indevassáveis para a votação e, é de entendimento comum que sistemas eleitorais que não façam uso de cabines indevassáveis, não dão garantia de inviolabilidade do voto e nem impedem a coação de eleitores.

A garantia do sigilo do autor do voto também tem que ser absoluta, pois basta existir uma mínima possibilidade teórica de violação do voto de um eleitor, que a coação de eleitores se implanta e produz seus efeitos deletérios sobre o resultado eleitoral, mesmo que os agentes coatores não consigam, de fato, identificar o autor de cada voto.

Diferente do sigilo telefônico e bancário, o conceito de sigilo absoluto do autor do voto impõe que nem mesmo juízes possam ordenar a sua quebra.

Entre as consequências do princípio da inviolabilidade absoluta do voto, se tem:

- *A identificação do eleitor não deveria ser feita no mesmo equipamento eletrônico no qual o eleitor vota, para evitar que algum software malicioso possa vincular de forma sistemática o conteúdo do voto com a identidade do seu autor;*
- *O registro e o processamento digital do voto não deveriam incluir ou reter dados de rastreamento que remetam à origem do voto, tais como, identificação da máquina de origem, o momento exato da votação, a identificação de quem estava "logado" no equipamento, etc.*

A necessidade de manter total irrastreabilidade do voto digital (inviolabilidade absoluta) associada ao direito do cidadão comum de poder entender o processamento do seu voto (soberania do cidadão), são características que tornam o processamento digital do voto uma atividade com peculiaridades únicas que, por exemplo, resultam ser muito mais difícil processar de forma segura e transparente um voto digital do que transferir uma enorme quantia de dinheiro por computadores.

2.5. Princípio da Publicidade e o Voto Digital

Como já dito acima, em 2009, a Corte Constitucional da Alemanha estabeleceu jusrisprudência sobre a transparência e a confiabilidade de sistemas eleitorais informatizados ao decidir que tais sistemas devem atender ao Princípio da Publicidade, de maneira que o eleitor comum (sem conhecimentos especiais de informática) tenha o direito de poder ver, compreender e conferir o registro e o destino do seu voto, e os candidatos tenham o direito de poder conferir, também com recursos próprios, a contagem eletrônica dos votos.

E, nessa toada, declarou inconstitucionais Sistemas Eleitorais de 1ª Geração – aqueles com gravação digital direta do voto, como ocorre nas urnas eletrônicas brasileiras – que não permitem ao eleitor ver para quem foi registrado e será contado o seu voto, e nem aos candidatos conferir ou auditar a apuração voto a voto, como se pode ver neste outro trecho da tradução apresentada pelo CMind [CMind 2010]:

"Decisão

2. A utilização de máquinas de votar Nedap ESD1 e ESD2 (máquinas DRE sem voto impresso conferível pelo eleitor) na eleição do 16º Parlamento Alemão não estava de acordo com o PRINCÍPIO DE PUBLICIDADE no processo eleitoral implícito no artigo 38, conjugado ao artigo 20, parágrafos 1 e 2 da Constituição.

Fundamento 111

O PRINCÍPIO DA PUBLICIDADE exige que todos os passos essenciais da eleição estejam sujeitos à comprovação pública. A contagem dos votos é de particular importância no controle das eleições.

Fundamento 155

Os votos foram registrados somente em memória eletrônica. Nem os eleitores, nem a junta eleitoral ou os representantes dos partidos poderiam verificar se os votos foram registrados corretamente pelas máquinas de votar. Com base no indicador no painel de controle, o mesário só pode detectar se a máquina de votar registrou um voto, mas não se os votos foram registrados sem alteração. As máquinas de votar não previam a possibilidade de um registro do voto independente da memória eletrônica, que permitisse aos eleitores uma conferência dos seus votos"

Mas, destaque-se que o Princípio da Inviolabilidade Absoluta do Voto não conflita com o Princípio da Publicidade no processo eleitoral, pois é o CONTEÚDO DO VOTO que deve ser PÚBLICO e conferível pelo eleitor (no ato de votação) e pelo fiscal (no ato de apuração), enquanto é o AUTOR DO VOTO que deve ser SECRETO a todo instante.

A adoção da transparência em todos os procedimentos no processo eleitoral eletrônico, incluindo a abertura do software para validação e certificação prévias, é chamado de Modelo de Segurança por Transparência.

Em oposição a esse modelo aberto, há o Modelo de Segurança por Ofuscamento, onde os procedimentos de registro dos votos e o software usado são fechados e ficam sob controle absoluto da equipe de desenvolvedores e operadores do sistema. Nesse modelo fechado, que protege a segurança do administrador, em detrimento do direito do eleitor e do candidato ao Princípio da Publicidade, se pretende que a eventual confiança subjetiva dada aos administradores seja aceita como garantia indireta da confiança técnica sem, no entanto, permitir a determinação efetiva desta.

2.6. Princípio da Independência do Software

O Princípio da Independência do Software em Sistemas Eleitorais estabelece que:

"Um sistema eleitoral é independente do software se um erro não-detectado no software não puder causar um erro indetectável no resultado da apuração ou na inviolabilidade do voto"

Foi criado, e se tornou necessário, a partir da constatação prática de que é muito mais difícil e caro se determinar que um software eleitoral complexo está livre de erros que afetem o seu desempenho, do que desenvolver esse próprio software.

Sobre a dificuldade e quase impossibilidade de se estabelecer confiabilidade técnica do software de sistemas eleitorais destacamos os seguintes pareceres de destacados autores internacionais na área de segurança em TI, conforme tradução nossa:

- Ph.D. Ronald R. Rivest (MIT - criador da técnica de criptografia assimétrica RSA e da Assinatura Digital) e Jonh Wack (NIST) [Rivest 2006]

"A tarefa de encontrar todos os erros num grande sistema é geralmente considerada impossível ou extremamente cara. Nossa habilidade para desenvolver software, de longe supera nossa habilidade de provar seu funcionamento correto ou de testá-lo satisfatoriamente dentro de restrições econômicas razoáveis (um teste completo de confiabilidade num software eleitoral certamente terá custo proibitivo).

Um sistema eleitoral no qual a integridade do resultado depende do funcionamento correto do seu software sempre será, de alguma forma, suspeito e irá requerer verificações sistemáticas do software, mesmo depois de uma completa (e cara) certificação por normas federais."

- Ph.D. Richard M. Stalmann (MIT - criador do projeto do software livre) [Stallman 2008]

"Votar com computadores é uma grande porta aberta para a fraude. O computador executa um software, e o software pode ser alterado ou substituído. Ele pode ser substituído apenas temporariamente por outro, durante a eleição, projetado para dar falsos totais. Nenhum estudo do software que deveria ser executado pode assegurar que o programa efetivamente executado não age mal.

O voto eletrônico é um evento especial, porque, normalmente, o eleitor não consegue estar atento a todas as partes envolvidas e descobrir se o seu voto foi corretamente contado. Não podemos assumir que o fabricante é honesto, ou que a autoridade eleitoral é honesta ou que os dois não conspiram juntos. Um sistema eleitoral deve ser a prova de tudo isso, mas isso é impossível num sistema puramente computacional."

O que há de comum nos pareceres desses dois autores é que determinar a confiabilidade técnica de sistemas eleitorais eletrônicos por meio da validação e certificação do software de fato utilizado é tarefa muito complexa que exige muitos procedimentos de alto grau tecnológico antes e durante a votação e apuração, proibitivamente cara e, na prática, impossível dentro de condições razoáveis de tempo e de orçamento.

Conforme as Diretrizes Voluntary Voting System Guidelines (VVSG) [Nist 2009], sistemas eleitorais devem oferecer total verificabilidade do resultado por via independente do software usado, estabelecendo as seguintes recomendações, assim traduzidas e adaptadas:

- *Ao menos dois registros do voto devem ser produzidos e um deles deve ser guardado em meio que não possa ser modificado pelo sistema (eletrônico) de votação, de forma que ambos registros não estejam sob controle de um único processo digital;*

- *O eleitor deve estar capacitado para verificar a igualdade dos dois registros do seu voto antes de deixar o local de votação;*
- *O processo de verificação dos registros do voto devem ser independentes e ao menos um deles deve ser conferível diretamente pelo eleitor;*
- *Os dois registros de um voto poderão ter sua consistência verificada posteriormente por meio de identificadores únicos que permitam a correlação dos registros.*

2.7. Disponibilidade Absoluta

Uma eleição não pode ser parcialmente adiada por indisponibilidade do sistema computacional. Deve ocorrer na data e hora marcadas em todo o território, não podendo ser postergada, nem mesmo parcialmente, por falha de qualquer origem, inclusive por ataque externo.

Assim, torna-se necessária forte resistência a ataques do tipo DoS (Denail of Service).

2.8. Outros Conceitos Desejáveis

Além dos conceitos essenciais e necessários acima descritos, interessa que outros conceitos também sejam atendidos por um sistema eleitoral digital como:

- Usabilidade - sistema amigável ao eleitor
- Direito de Refutação (dentro do local de votação)
- Celeridade – na votação e na apuração
- Recuperação de erros e retomada desburocratizada
- Portabilidade e logística econômicas
- Treinamento sempre disponível para eleitor e mesário
- Distribuição matricial de equipamentos e mesas eleitorais

3. Tabela de Conformidade

A tabela seguinte descreve a conformidade de três modelos de máquinas de votação com relação a conceitos derivados dos requisitos apresentados acima.

Os equipamentos analisados são os seguintes:

- Urnas Eletrônicas brasileiras, desenvolvidas pelo TSE, em uso no Brasil desde 1996. Também foram usadas no Paraguai, na Argentina e no Equador em 2004/6 mas depois foram abandonadas nesses países. Classificadas como de 1^a geração segundo [Rezende 2010]
- Equipamentos de votação SAES 3000, fabricadas pela empresa Smartmatic e usadas desde 2004 na Venezuela e mais recentemente na Bélgica e no Equador. Classificadas como de 2^a geração segundo [Rezende 2010]

- Equipamentos Vot-AR, fabricadas pela empresa MSA e usadas em algumas províncias da Argentina desde 2010 e mais recentemente no Equador. Classificadas como de 3ª geração segundo [Rezende 2010]

Tabela 2. Tabela de Conformidade

	UE2009 Brasil	**SAES Venezuela**	**Vot-Ar Argentina**
Princípio da Publicidade			
Gera voto impresso conferível pelo eleitor	NÃO	SIM	SIM
Eleitor pode conferir o conteúdo da gravação digital do voto antes de sair do local de votação	NÃO	NÃO	SIM
Fiscal externo pode verificar a igualdade entre os diversos registros do voto	-	NÃO	SIM
Fiscal externo pode acompanhar e verificar a contagem dos votos de cada seção eleitoral	NÃO	SIM	SIM
Princípio da Inviolabilidade Absoluta do Voto			
Garantia contra a violação do voto causada por um erro não detectado no software	NÃO	SIM	SIM
Garantia contra a violação do voto causada por reordenação do arquivo dos votos	NÃO	NÃO	SIM
Princípio da Independência do Software			
Uma modificação ou erro não detectado no software pode causar um erro indetectável no resultado da apuração	SIM	NÃO	NÃO
Conformidade com a Norma Técnica: *Voluntary Voting System Guidelines* (Seção 7.8)	NÃO	SIM	SIM
Outros Conceitos Desejáveis			
Tempo para publicação na Internet dos resultados por Seção, para fiscalização da Totalização	72h (2012)	?	2h (2011)
Conferência da assinatura digital do software feita em equipamento sob controle do fiscal	NÃO	?	SIM
Solução simples de diferenças entre o registro no papel e o registro digital do voto	-	NÃO	SIM
Distribuição matricial de urnas e mesas: o eleitor pode escolher uma urna livre para votar, sem ter	NÃO	SIM	SIM

que esperar que um eleitor anterior complete seu voto. Menores filas.			
Eleitor pode escolher a ordem dos cargos a votar	NÃO	SIM	SIM
Adaptação para plebiscitos e outras consultas - disponibilidade de opções "sim", "não", ou outras mais específicas	NÃO	SIM	SIM
Preparação simplificada, sem introdução de dados diferentes para cada seção/máquina	NÃO	NÃO	SIM
Troca de equipamento defeituoso e necessidade de recuperação de dados	LENTA SIM	LENTA SIM	RÁPIDO NÃO

4. Conclusões

Para finalizar, excluído o requisito da Disponibilidade Total, o processo eleitoral eletrônico brasileiro, com o respectivo equipamento de votação (urnas eletrônicas), não consegue atender aos demais requisitos e conceitos estabelecidos no presente trabalho, tais como: Soberania dos Direitos dos Eleitores e dos Candidatos, Tripartição dos Poderes, Princípio da Publicidade, Princípio da Inviolabilidade Absoluta do Voto e Princípio da Independência do Software (ou da Verificabilidade do Resultado Independente do Software).

A substituição da auditoria simplificada, pelos eleitores e candidatos, por uma validação e certificação exaustiva do software por agentes do Ministério Público, da OAB e dos partidos políticos foi totalmente ineficaz.

Dessa forma, o processo eleitoral eletrônico brasileiro recai a um processo sem garantias concretas de inviolabilidade e de justa apuração, onde se torna muito difícil estabelecer a sua confiabilidade técnica.

E essa dificuldade prática para estabelecer a confiabilidade técnica do software efetivamente usado no dia da eleição no Brasil, acrescida da não previsão de uma auditoria contábil deveras independente (pelos candidatos), equivale à adoção do Modelo da Segurança por Ofuscamento, descrito acima, e pretende que a eventual confiança subjetiva dada aos administradores eleitorais seja tomada como garantia indireta da confiança técnica sem, no entanto, permitir a determinação efetiva desta.

Vale lembrar que a confiabilidade subjetiva não pode ser imposta. Deveria ser conquistada, e a melhor forma de conquistá-la seria permitindo que se possa determinar a confiabilidade técnica objetiva.

Referências

Neumann, P.G. - Security Criteria for Electronic Voting - 16th National Computer Security Conference Baltimore, Maryland, September 20-23, 1993. - http://www.csl.sri.com/users/neumann/ncs93.html. Acessado em 07/09/14.

CT, Computer Technologists' Statement on Internet Voting – disponível em: http://www.verifiedvoting.org/wp-content/uploads/2012/09/InternetVotingStatement.pdf 2012

Pieters, W. - Verifiability of electronic voting: between confidence and trust. In: Data Protection in a Profiled World. Springer, Dordrecht, pp. 157-175. ISBN 9789048188642, 2010. - http://doc.utwente.nl/72498/

Três, C.A.- A Soberania do Povo na Fiscalização do Exercício de sua Soberania. In: Seminário do Voto Eletrônico, Câmara dos Deputados, 29 de maio de 2002. - http://www.brunazo.eng.br/voto-e/textos/tres2.htm

TCFA, Decisão original do Tribunal Constitucional Federal da Alemanha em 03/03/2009
http://www.bundesverfassungsgericht.de/entscheidungen/cs20090303_2bvc000307.html

CMind, Comitê Multidisciplinar Independente. Relatório sobre o Sistema Brasileiro de Votação Eletrônica. Brasília: Edição dos Autores, 2010. Os temas referidos se encontram nas Seções 4.1.1 e 4.1.2- http://www.votoseguro.org/textos/CMind-1-Brasil-2010.pdf

Aranha, D. et al. - Vulnerabilidades no software da urna eletrônica brasileira. UnB, 2012. 36 pp. - disponível em http://sites.google.com/site/dfaranha/projects/relatorio-urna.pdf, visitado em 24 de setembro de 2014

Rivest R.R. , Wack, J.P. - On the notion of "software independence" in voting systems : USA, NIST/MIT, 28/07/2006 - http://people.csail.mit.edu/rivest/pubs/RW06.pdf

Stallman – A Opinião de Richar Stallman - http://www.vialibre.org.ar/2008/11/12/voto-digital-la-opinion-de-richard-stallman/

NIST, Voluntary Voting System Guidelines - NIST/US-EAC (2009). - Definição de Independent Verification Systems na Seção 7.8 - http://www.eac.gov/assets/1/AssetManager/VVSG_Version_1-1_Volume_1_-_20090527.pdf

Rezende, P. D. - Votação Eletrônica, 3ª Geração. CMind, 2010 – apresentado em cerimônia pública no TSE - http://www.cic.unb.br/docentes/pedro/trabs/TSE3G.pdf

Proposta de um modelo de auditoria para o sistema brasileiro de votação utilizando criptografia visual

Carlos Eduardo, Gleudson P. V. Junior, Wagner M. Santos, Ruy J. G. B. de Queiroz,

Centro de Informática - Universidade Federal de Pernambuco (UFPE)
Recife – PE - Brazil

{cers, gpvj2, wms2, ruy}@cin.ufpe.br

Abstract **–** The paper aims to the introduction of improvements in the Brazilian electronic ballot, introducing the possibility of individual verifiability and audit of votes, without the loss of confidentiality of such. For that purpose is presented a voting protocol using visual cryptography already proposed by David Chaum.
Keywords: Visual Criptography, voting systems, auditing, vote.

Resumo **–** O artigo objetiva a introdução de melhorias na urna eletrônica brasileira, introduzindo a possibilidade da verificabilidade individual e auditoria dos votos, sem que ocorra a perda do sigilo dos mesmos. Para isso, é apresentado um protocolo de votação utilizando criptografia visual já proposto por David Chaum.
Palavras-chaves: Criptografia visual, sistemas de votação, auditoria, voto.

1. Introdução

O Brasil é considerado um país democrático que tem participação indireta do povo, exercendo assim a democracia através dos seus representantes. Para a escolha destes se faz necessário a utilização de um processo que garanta, de forma justa, uma eleição segura.

Com a informatização das eleições em 1996 foi apresentado ao mundo a aplicabilidade de um modelo de votação, que segundo autoridades brasileiras, é apontado como 100% seguro contra fraudes. Esse processo eleitoral informatizado se disseminou pelo país ao poucos, até que no ano 2000 todos os municípios brasileiros utilizaram tal processo.

Para muitos o sistema de votação eletrônico aponta um grande avanço no país, afinal todo o processo de montagem das urnas ocorre no Brasil e o resultado final de uma eleição é dado com muita rapidez. Por outro lado, houve quem discordasse da segurança do processo. Ao longo destes anos de utilização das urnas eletrônicas, ocorreram diversas acusações de fraudes e muitos relatórios com provas circunstâncias foram apresentados, colocando assim em cheque a segurança das urnas (Cunha, et al., 2010).

O objetivo deste trabalho é propor um novo modelo utilizando um esquema com criptografia visual, a fim de prover possíveis mecanismos que atendam um processo de auditoria e verificabilidade individual com a materialização do voto de um modo não tradicional, tendo como foco o emprego desse esquema no sistema brasileiro de votação. Este esquema é baseado em (Chaum, 2002) e (D. Chaum, 2007).

A principal fraqueza do protocolo proposto é a necessidade de que parte do número de eleitores retornem ao local de votação depois para ocorrer a verificabilidade individual e auditoria dos votos, contudo isto reforça a segurança do mecanismo, discutiremos isto mais adiante.

O artigo está organizado da seguinte forma: na seção 2 serão abordadas as principais tecnologias utilizadas em sistemas de votação, desde um sistema de votação de primeira e segunda geração, até chegarmos ao modelo mais recente: as maquinas de terceira geração. Na seção 3 serão apresentadas as definições de criptografia visual, mecanismo que é apresentado como uma parte da solução a ser implantada no processo eleitoral. Na seção 4 será apresentada uma proposta de um modelo com criptografia visual sendo aplicado ao sistema de votação brasileiro, como forma de provê um processo de auditoria. Por fim, serão apresentadas as conclusões.

2. Tecnologias para sistemas de votação

Atualmente existem diversos sistemas de votação em uso no mundo. Nesta seção abordaremos alguns dos modelos mais conhecidos, classificados como: sistemas de votação de 1ª geração, de 2ª geração e 3ª geração.

Máquinas de 1ª geração

Conhecidas como máquinas de gravação eletrônica direta, do inglês *DRE (Direct Recording Electronic)* são dispositivos que dependem completamente do software para o registro do voto, de tal modo que não possuem outras formas de comprovação do mesmo, como por exemplo, a impressão. Seu emprego foi iniciado amplamente na década de 90, em países como Holanda, Índia, Alemanha e Brasil. Este último foi o primeiro a realizar uma eleição 100% informatizada, ou seja, do alistamento até a apuração e totalização dos votos usando urnas eletrônicas DRE (Monteiro, Soares, Oliveira, & Antunes, 2001).

Uma DRE é um equipamento composto por hardware e software, no qual o eleitor registra seu voto através do toque em tela ou ainda através de um teclado especial, como ocorre nas eleições brasileiras. As DREs armazenam os votos em um dispositivo de memória, podendo ou não estar cifrados. Ao fim do sufrágio os votos são encaminhados para o local autorizado pelo órgão competente e então é feita a totalização dos votos (Costa, 2008). A Figura 1 (Rezende, 2010) ilustra a custódia do voto em maquina deste tipo.

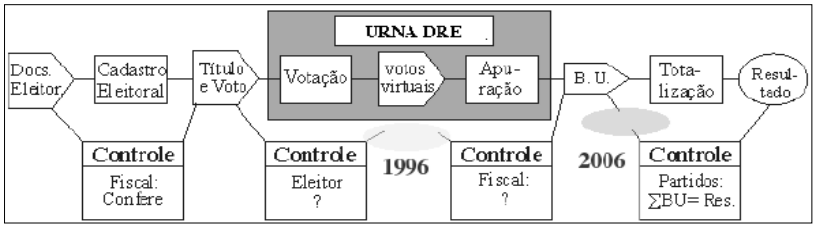


Figura 1 - Custódia do voto em máquinas de 1ª geração (Rezende, 2010)

Vantagens:

- A totalização dos votos ocorre de forma quase automática.
- Não há como existir rasuras ou má marcação por parte do eleitor.
- Possui boa interface e um design simples, o que facilita o seu uso por diversas pessoas.
- São projetadas para suportar diversos ambientes.

Desvantagens:

- Falsa ideia de segurança absoluta.
- Extremamente difícil de realizar a detecção de uma fraude.
- Por não existir a materialização do voto, não é possível realizar uma recontagem dos mesmos.
- Descarta a possibilidade de auditoria por parte do eleitor.
- Os eleitores não sabem como funcionam as DREs, no tocante a programação da máquina, fazendo com que seja necessário confiar plenamente nos projetistas e órgãos responsáveis pelo desenvolvimento

Máquinas de 2ª geração

Ronald L. Rivest e John P. Wack foram os primeiros a propor o conceito do princípio da independência do software (Rivest & Wack, 2006), onde se uma modificação ou erro não detectado ocorrer no software, este não poderá causar uma modificação ou erro indetectável no resultado da apuração dos votos. As máquinas que utilizam este princípio são chamadas de maquinas de votar de 2ª geração, pois conferem uma segunda maneira de contabilizar os votos sem a dependência total do registro eletrônico.

Exemplos destas máquinas são os sistemas de contagens por scanner ópticos, do inglês *PCOS (Precint Count Optical Scan)* e DREs com o voto impresso conferível pelo eleitor, do inglês *VVPAT (Voter Verified Paper Audit Trails)*. A Figura 2 (Rezende, 2010) ilustra a custódia do voto em maquina do tipo VVPT.

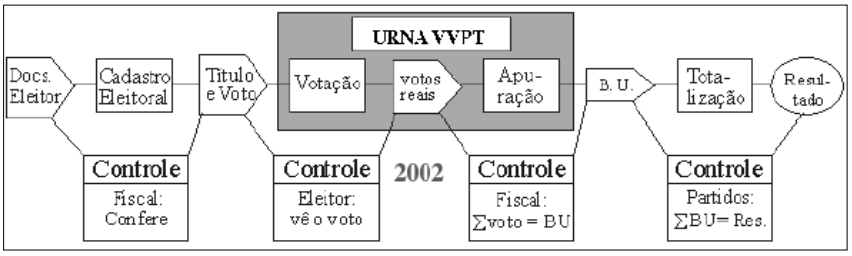


Figura 2 - Custódia do voto em máquinas de 2ª geração (Rezende, 2010)

Vantagens:

- Permite processo de auditoria futuras, visto que a cédula marcada pelo eleitor já consiste em si mesma a materialização do voto.
- Com o voto depositado no scanner, o eleitor sabe se o mesmo foi realmente registrado, visto que esse tipo de máquina rejeita cédulas com marcações incorretas, dando ainda a chance de o eleitor utilizar outra em branco.

- Em caso de falta de energia a votação poderá continuar, pois a maioria dos escâneres possui bateria interna.

Desvantagens:

- Dificuldade com acessibilidade para eleição com muitos candidatos.
- Pode ocorrer manipulação no resultado e mesmo assim a fraude passar despercebida.

Máquinas de 3ª geração

Também conhecidas como *E2E* (*End-to-End voter verifiable*), esses sistemas de votação são dotados de características que atendem integridade e sigilo do voto. O seu projeto tem por base permitir a verificação do voto pelo eleitor, fornecendo um mecanismo que possibilita saber se os votos não foram modificados, ao mesmo passo que não revela em quais os candidatos o eleitor votou. Da mesma maneira que as máquinas VVPT, os sistemas de 3ª geração possuem duas vias de verificação, o software e a cédula, no caso com o pequeno diferencial de o eleitor poder receber um recibo de seu voto. Com esse processo, disponibiliza-se uma verificação fim-a-fim do voto, o que garante que o voto marcado é o voto computado. Este recibo insere outra característica importante, a possibilidade de se identificar em um processo de auditoria qual das duas vias foi possivelmente adulterada, o que não ocorre nas maquinas de 2ª geração.

Uma das principais distinções está na cédula, onde o sistema E2E emprega um código de verificação. Esse código de verificação será fornecido ao eleitor como recibo, possibilitando ao mesmo verificar se o seu voto está de acordo com a sua intenção, sem, entretanto, fazer qualquer tipo de vínculo direto a intenção do voto a um candidato específico. Dessa forma qualquer eleitor ou terceira parte pode verificar o voto, no entanto não saberá em quem foi votado, garantindo assim a propriedade do sigilo.

As características deste sistema possibilita diversas maneiras de implementação, a grande maioria delas apresentando baixo custo. A votação pode ser realizada com papel e caneta pura e simplesmente, sem fazer o emprego de qualquer artifício criptográfico e talvez obscuro para a maioria dos eleitores leigos. Entretanto, o emprego da criptografia é expressamente recomendado nesse modelo, visto que a automação além de agilizar o processo como um todo, também adiciona maior confiabilidade e segurança.

3. Criptografia visual

Em 1994, os israelitas Noni Naor e Adi Shamir foram os primeiros a proporem a criptografia visual. Neste esquema não existe a necessidade de um computador para decriptar a mensagem, ao invés disto, é suficiente o sistema visual humano. No modelo utiliza-se uma página impressa com texto cifrado e outra impressa em transparência (que serve como chave), a decriptação ocorre quando as duas páginas são sobrepostas, de maneira a revelar o texto puro. Embora existam ruídos sobre o resultado, qualquer pessoa pode entender a mensagem (Naor & Shamir, 1994).

Tendo em vista a descrição acima, podemos definir criptografia visual como uma técnica especial de encriptação, utilizada para esconder a informação em imagens, ou mesmo textos transformados em imagens, de tal maneira que ela pode ser

decodificada pelo sistema visual humano, se a imagem correta (a chave) for usada. No esquema proposto pelos autores, as duas imagens (camadas) são necessárias para revelar a informação, ou seja, sem uma delas é impraticável obter o conteúdo original. A Figura 3 mostra um exemplo básico de criptografia visual.

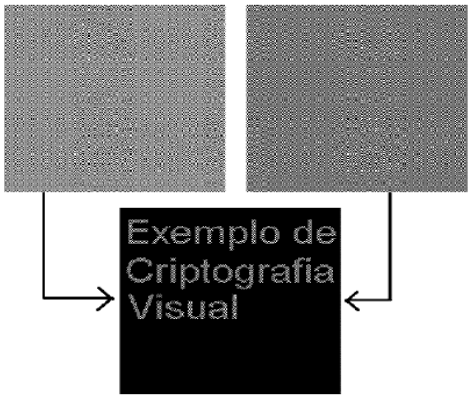


Figura 3 - Exemplo de criptografia visual (Monteiro, Soares, Oliveira, & Antunes, 2001)

Criptografia visual em eleições

Em 2002, David Chaum foi o primeiro pesquisador a sugerir o emprego de criptografia visual em eleições (Jones, 2010). O sistema proposto permite a verificabilidade do voto por parte do eleitor e mesmo oferecendo um recibo do voto, mantém o seu sigilo.

O autor apresenta o sistema da seguinte forma:

1) O eleitor introduz a sua escolha através de uma tela de toque ou outro meio de entrada, é feita então uma impressão (parte do qual será o seu recibo). Nela é listado o nome do candidato escolhido junto com a filiação partidária e o cargo que este concorre.
2) Depois de imprimir o voto, a máquina pede para rever a impressão ainda na impressora e aceitá-lo ou não, dando a oportunidade de alterá-la e gerar uma nova impressão.
3) Com a concordância do voto, a máquina pede para o eleitor apontar qual das partes: superior ou inferior da camada será mantida como recibo. Isto é necessário, pois a impressão é feita com as duas camadas juntas e alinhadas. Após indicar a escolha, a impressora imprime a parte final do recibo, ou seja, o voto criptografado usando criptografia visual. Esta ultima parte é diferente porque possui as mensagens que podem ser lidas depois que as camadas são separadas.
4) O eleitor toma as duas camadas juntas e as separa. A parte escolhida anteriormente como recibo pode conter a mensagem legível impressa, por exemplo: “Eleitor mantenha esta camada do recibo privada e protegida”, enquanto que a outra pode ter: “Eleitor entregar esta camada para a autoridade eleitoral”.
5) Antes de sair do local de votação, é entregue a autoridade eleitoral uma parte para ser destruída por um triturador de papel e a outra é levada para casa. O voto fica armazenado eletronicamente na maquina e será enviado pela Internet a um site para verificações posteriores. Os bits na camada de

papel triturado também são "picados" eletronicamente – isto é, as únicas coisas que permanecem do voto são a camada física e, na máquina, uma versão digital da mesma imagem.

6) Quando termina a votação, apenas os votos digitais são enviados, por meio eletrônico ou por outra mídia de armazenamento.
7) Após um tempo pré-estabelecido da votação, o eleitor pode verificar o seu voto na pagina da web do sistema em questão. Para isto ele utiliza o numero de identificação contido na camada que ficou como recibo.

Em (D. Chaum, 2007) foi proposto um modelo semelhante ao descrito há pouco usando criptografia visual para conseguir verificação individual e auditoria via internet. Em (Graaf, 2004) existe uma abordagem semelhante com foco no uso no sistema de votação brasileiro, contudo, sem a utilização de criptografia visual.

Autenticação em votação remota

Em abril de 2003, em (Paul, Evans, Rubin, & Wallach, 2003) é apresentado um sistema de votação via Internet utilizando criptografia visual para autenticar o eleitor.

No sistema em questão, as autoridades eleitorais geram as transparências cifradas e enviam-nas junto com uma lista de endereços de eleitores para um terceiro (os correios), que envia a cada eleitor uma transparência selecionada aleatoriamente, juntamente com um pacote de informações, incluindo instruções de voto e uma senha.

Com o pacote em mãos o eleitor acessa a pagina da web e digita a senha que recebeu, o sistema irá solicitar informações pessoais, após a confirmação dos dados é apresentada na tela a *share* (a outra parte da transparência). O eleitor então coloca a sua transparência sobre a tela de modo a sobrepô-las, deste modo a nova senha é revelada. Depois de utilizá-la o eleitor poderá votar. A Figura 4 mostra a operação de sobreposição das transparências.

Figura 4 - Sobreposição da transparência em tela (Paul, Evans, Rubin, & Wallach, 2003)

O protocolo serve para autenticar o eleitor, para que o mesmo tenha garantias que o site utilizado é o oficial e a transparência será o seu comprovante físico. Segundo os próprios autores, a implementação não aumenta de forma drástica o custo para realização de uma eleição.

4. Modelo proposto

David Chaum foi o pioneiro em sugerir a aplicação da criptografia visual em eleições (Chaum, 2002) como já mencionado anteriormente. Esta tecnologia pode ser implantada no sistema eletrônico de votação brasileiro conservando quase que todo o modelo existente, adicionando apenas algumas como alterar o software e o acréscimo de uma impressora para poder imprimir as transparências que servirão como recibos para os eleitores. As mudanças ocorreriam somente na fase de votação, acrescentando um processo de auditoria.

As cédulas seriam semelhantes às sugeridas por David Chaum em (Chaum, 2002), ou seja, teriam uma parte transparente, com as opções do candidato escolhido, encriptadas, usando técnicas de criptografia visual e um número de identificação, que será utilizado posteriormente para a auditoria do processo. Além disso, existiria uma região em branco para receber a marca oficial como forma de evitar possíveis fraudes. A Figura 5 ilustra um exemplo da cédula do modelo proposto.

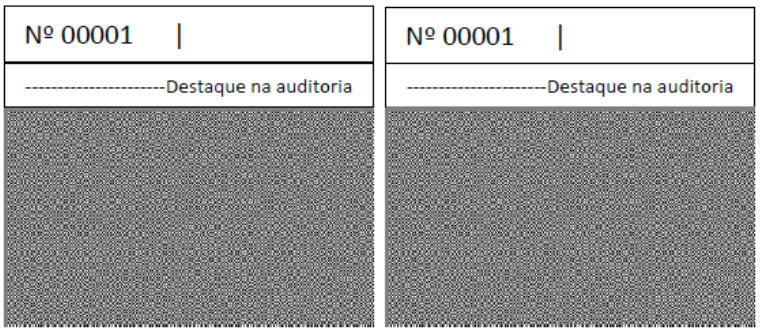


Figura 5 - Partes separadas da cédula utilizando criptografia visual.

O processo de votação deve seguir as seguintes etapas:

1) O eleitor continuará utilizando a urna eletrônica, procedendo normalmente na escolha dos candidatos de sua preferência;
2) Existirá a possibilidade de imprimir ou não as duas cédulas cifradas. Depois de impressas, o eleitor confere se o voto foi devidamente impresso sobrepondo-as, em caso negativo ele poderá cancelar o voto e reiniciar o processo. Em caso positivo vai para a próxima etapa;
3) Ele deposita uma das cédulas em uma urna tradicional e a outra é entregue ao mesário para receber um carimbo (uma marca oficial que dificulte fraudes) no local em branco da cédula (Figura 5, parte superior direita), em seguida, o mesário devolve a cédula que servirá como comprovante em uma futura auditoria do processo.

A Figura 6 mostra a arquitetura geral do modelo proposto:

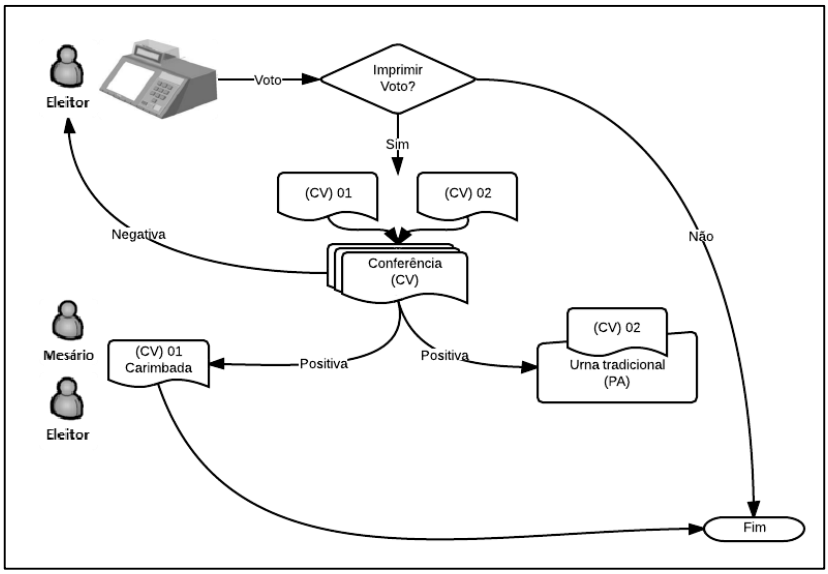


Figura 6 - Arquitetura geral do modelo proposto

A impressão na transparência é totalmente viável, visto que a urna eletrônica possui entrada para impressoras e empresas como a HP, por exemplo, já fabricam modelos suficientemente capazes de imprimi-las fazendo uso da criptografia visual. Outra possibilidade para impressão é utilizar uma impressora especial como foi sugerido em (Chaum, 2002), no entanto neste caso os gastos são maiores. A Figura 7 mostra a concepção física do modelo.

Figura 7 - Concepção física do modelo proposto

As urnas devem ser mantidas organizadas por seção e lacradas até o dia da auditoria, visto que as cédulas serão reutilizadas. Não existe nenhum risco à privacidade dos votos caso uma pessoa veja uma das cédulas, pois o voto está encriptado.

O sistema de criptografia pode ainda empregar o método *Content Area and Black Area Method* (um esquema visual 2 por 2) (Naor & Pinkas, 1997) e, para reforçar mais à segurança, pode-se utilizar algum método de criptografia visual de autenticação como visto na seção 3.

No caso de empregar o mecanismo de autenticação, uma sugestão é dividir a cédula em nove áreas, que servirá para escolher em qual delas será impressa o voto. O

eleitor digita um número variável entre 1 e 9 no teclado da urna que seja correspondente ao espaço na cédula. A Figura 8 mostra o uso da autenticação na urna.

Figura 8 - Autenticação na urna brasileira

Com este método de autenticação a cédula ficaria preenchida apenas na região correspondente a região escolhida e o restante seria apenas região de ruído, ou seja, uma área com preenchimento sem valor. Como só eleitor sabe qual foi a área que escolheu, dificulta a fraude, pois haveria quem estaria tentando corromper a cédula teria quase 90% de errar o local escolhido.

A auditoria ocorre dias após as eleições, em um tempo pré-determinado pelo TSE. No dia correspondente, uma empresa deve ser contratada para conferir se o lacre não foi rompido anteriormente e caso esteja tudo correto o mesmo é rompido na presença de terceiros. Os votos são retirados das urnas de acordo com cada seção, colocados em ordem numérica e posteriormente são entregues aos eleitores; isto é possível porque as cédulas possuem um numero de identificação (Figura 5). Os eleitores vão ao local devido portando a cédula (comprovante), lá são devidamente identificados pelos oficiais eleitorais, que entregam a cédula outrora depositada na urna para serem auditadas pelo eleitor; este vai a um local reservado para ninguém ver a sua opção e não quebrar o sigilo do seu voto, lá sobrepõe as duas cédulas; desta maneira o voto será revelado, o eleitor confere o mesmo, remove em ambas as cédulas o numero de identificação e os conecta de alguma forma (grampo, cola etc.); em seguida o eleitor deposita em uma urna para ser auditada pelos oficiais eleitorais que farão isto em outro momento na presença de terceiros.

A auditoria sobre a cédula cifrada com criptografia visual é diferente da auditoria sobre a cédula tradicional, pois aqui primeiro atribuímos ao eleitor o poder de verificar depois de algum tempo que seu voto é computado corretamente. Isto ocorre devido a cédula está divida em duas partes que se completam. Quando o eleitor volta ao local de votação ele tem a oportunidade de tomar em suas mãos a outra parte da cédula e verificar que o voto está registrado, remove o código identificado para que não seja quebrado o sigilo do voto e só então é que ele conecta ambas as partes tornando assim a cédula cifrada semelhante à cédula tradicional para ser depositada novamente a urna, para ser contada. Aqui um passo é extremante necessário para que fraudes não ocorram, como ocorreram ao longo da história, os votos presentes nas urnas podem ser recontados em um intervalo de tempo menor, podem-se colocar até mesmo câmeras no

local para vigiar estas urnas. A auditoria pode começar pela manhã e a tarde a recontagem do voto já poderá começar a ocorrer.

Vantagens do Modelo Proposto:

- Aumento da segurança do processo, fazendo com que o eleitor possa auditar seu voto e depois o TSE poderá fazer o mesmo.
- O eleitor não precisa se preocupar com a quebra do sigilo do voto, pois a cédula (comprovante) está criptografada de modo a não revelar o voto.
- Dificuldade em forjar uma cédula falsa, pois a original recebe uma marca oficial.
- Dificuldade em forjar uma nova cédula, devido ao uso da criptografia visual aplicada e por desconhecer qual a região foi escolhida pelo eleitor, caso algum método de autenticação for usado.

Desvantagens do Modelo Proposto:

- Aumento dos gastos com impressoras e dos dias necessários para a auditoria.
- Causa certa dificuldade para pessoas com necessidades especiais, idosas e outras.

No caso de haver contradições entre o resultado apresentado pelo voto eletrônico e o voto impresso o processo deve ser anulado e um novo ocorrerá.

Não é necessário que todos os eleitores produzam seus recibos, como ocorre com alguns sistemas de votação, uma amostragem significativa já seria suficiente para saber se houve fraude nas eleições, assim, embora pessoas com necessidades especiais, idosas e outras possam considerar o método complicado, elas podem ser dispensadas do processo. Contudo é importante que dos eleitores gerem os recibos para o caso de perdas, furtos ou comprovantes que precisem ser anulados. No Brasil um grande número de eleitores não se interessa pelo processo eleitoral, então estas também podem ser dispensadas, fazendo com que participem apenas as pessoas que realmente desejam uma auditoria séria.

5. Conclusão

A verdadeira democracia só pode ocorrer se o poder realmente emanar do povo, como está escrito na Constituição Federal. Para isto, é necessário ocorrer um processo eleitoral no qual a fraude não ocorra.

Este trabalho apresentou procedimentos em sistemas eleitorais que são empregados no mundo e no Brasil, agrupados em três gerações no intuito de mostrar como os países tentam conseguir uma eleição segura, entretanto todos esses sistemas possuem suas vantagens e desvantagens.

O Brasil se encaixa no grupo dos sistemas de votação que ainda utilizam máquinas do tipo DRE e embora este sistema forneça praticidade e rapidez no que tange à apuração dos votos, ele não possibilita a impressão do voto, que é uma solução para que a sociedade possa comprovar seu sufrágio utilizando um processo de auditoria. É neste momento que entra a criptografia visual como ferramenta importante que faz a ponte entre tecnologia e transparência, aumentando também a segurança.

Os profissionais e pesquisadores da área de segurança não acreditam que exista um sistema eleitoral livre da possibilidade de fraudes. Então porque utilizar a

criptografia visual? Podemos responder esta pergunta com outra: qual o motivo de se colocar portas resistentes e as melhores trancas em uma casa mesmo sabendo que um ladrão pode arrumar um meio de roubá-la? A resposta é simples: torná-la mais segura. O uso da criptografia visual não vai tornar o sistema inviolável, entretanto aumentará sua segurança e como vimos neste trabalho o seu uso é possível mesmo com o aumento dos gastos.

6. Referências

Chamon, O. (2008). *Direito Eleitoral.* São Paulo: Concursos Públicos.

Chaum, D. (2002). Secret-Ballot Receipts and Transparent Integrity. *Palo Alto Workshop on Information Dynamics in the Networked Economy.*

Costa, R. (2008). *Sistema Seguro de Votação Eletrônica Multi-Cédulas.* Curitiva: Pontificia Universidade Católica do Paraná.

Cunha, S., Marcacini, A., Cortiz, M., Fernandes, C., Stolfi, J., Rezende, P., et al. (2010). *Relatório sobre o Sistema Brasileiro de Votação Eletrônica.* Brasília: Editora dos Autores.

D. Chaum, J. v. (2007). Secret Ballot Elections with Unconditional Integrity. *Cryptology ePrint Archive, Report 2007/270*.

Graaf, J. v. (2004). Adapting Chaum's Voter-Verifiable election scheme to the. pp. 187-198.

Jones, D. (2010). On Optical Mark-Sense Scanning. Lowa: University of Lowa.

Marques, G., Vinicius, G., & Jorge, R. (2012). Impactos Computacionais de Uma Implementação de Criptografia Visual. *Revista Eletrônica de Sistema de Informação*.

Monteiro, A., Soares, L., Oliveira, R., & Antunes, P. (2001). *Sistemas Electrónicos de Votação.* Lisboa: Faculdade de Ciências da Universidade de Lisboa.

Naor, M., & Pinkas, B. (1997). Visual Authentication and Identification. *Crypto97*, pp. 322-336.

Naor, N., & Shamir, A. (1994). Visual Cryptography, Advances in Cryptology-Eurocrypt. *Lecture Notes in Computer Science*, pp. 1-12.

Paul, N., Evans, D., Rubin, A., & Wallach, D. (2003). Authentication for Remote Voting. *Workshop on Human Computer Interaction and Security System.*

Rezende, P. (Jul. de 2010). Votação Eletrônica, 3ª Geração. *Audiência Pública*.

Rivest, R., & Wack, J. (2006). On the notion of "software Independence" in voting systems. *National Institute of Standards and Technology (NIST)*.

Modelo Brasileiro de Votação Mecatrônica Independente de *Software* ou Votação Mecatrônica

Ronaldo Moises Nadaf

nadaf@nadaf.net

Resumo. *O artigo sugere a criação do "Modelo Brasileiro de Votação Mecatrônica Independente de Software" ou "Votação Mecatrônica," que propõe pesquisas e o desenvolvimento de novos equipamentos para um sistema automatizado de identificação do eleitor, coleta e apuração da votação, e de softwares capazes de gerar dois ou mais registros diferentes e auditáveis do voto digital e impresso.*

1.Introdução

Em 2009, o Congresso Nacional aprovou, e a Presidência da República sancionou, a lei 12.034 que previa, no art. 5º, a implantação do voto impresso e separação do terminal do eleitor da máquina de votar [1]. O objetivo era garantir a transparência, permitindo a auditoria dos votos e a implantação do conceito de sistema de votação independente de *software*[2], e também evitar que uma falha ou um erro viesse associar o voto coletado ao eleitor identificado no terminal.

Em 2011, o Ministério Público Federal moveu a ação de inconstitucionalidade nº 4543 contra o artigo 5º da lei e, em novembro de 2013, o Supremo Tribunal Federal acatou o pedido e declarou o artigo inconstitucional. A decisão, embasada num suposto reconhecimento mundial do sistema em uso, não levou em consideração pareceres técnicos e jurídicos emitidos por diferentes partes da ação [3].

A Suprema Corte também deixou de lado os resultados de um teste realizado pela própria autoridade eleitoral em 2012, quando uma equipe de cientistas revelou que o sistema brasileiro de votação é falho, defasado e carece de melhorias[4].

O objetivo desse artigo é apresentar uma proposta de evolução tecnológica do modelo atual de votação totalmente eletrônico para um modelo mecatrônico, ou seja, sugerir estudos para o desenvolvimento de dois novos equipamentos eletromecânicos que permitam a identificação biométrica do eleitor sem associa-lo à sua escolha, a impressão, a coleta, a apuração e armazenagem dos votos de maneira segura, operados por softwares distintos, capazes de gerar dois ou mais registros diferentes de voto, sendo um digital e outro impresso, e de realizar a auditoria do sufrágio sem que exista qualquer interferência humana no processo, onde apenas e tão o somente o eleitor tenha contato com o registro físico do voto.

Os modelos virtuais das máquinas aqui propostas, suas dimensões físicas e a cédula do voto, foi desenvolvida através de experiências realizadas em um ambiente

construído no software 3d Max Studio, onde os conceitos aqui descritos estão sendo colocados em prática em tamanho real e continuam em evolução.

2. Estrutura do Modelo de Votação Mecatrônica

No contexto tecnológico proposto, a primeira mudança antecede a votação, onde o próprio eleitor realiza a identificação biométrica das impressões digitais dos dedos e o reconhecimento da face. Em seguida vota em um teclado ou tela sensível ao toque, visualiza a foto do candidato e confirma o voto como já de costume.

A segunda mudança está na inclusão da "Mesa Receptora Independente", que serve de suporte para a Urna Eletrônica (ou tablet) e os leitores biométricos, além de realizar a impressão, a coleta ou a trituração das cédulas de votação.

Figura 1 - À esquerda, Mesa Receptora Independente e à direita, Máquina Apuradora

A Mesa Receptora tem acopladas duas caixas, uma chamada de "Urna Independente de *Software*", onde as cédulas válidas são armazenadas aleatoriamente; e outra, chamada de "Urna Ecológica", onde as cédulas inválidas trituradas são armazenadas em forma de resíduo. Após o encerramento da votação, a Urna Independente é retirada da mesa e inserida na "Máquina Apuradora de votos", um segundo equipamento que realiza o escrutínio e auditoria das cédulas, que confirmará o resultado apurado pela votação eletrônica.

Abaixo estão relacionados todos os equipamentos que compõem o modelo proposto:

1. Leitores biométricos das impressões digitais e de reconhecimento da face;
2. Cédula de Segurança em papel com três registros do voto do eleitor;
3. Dispositivo de registro digital do voto, que pode ser um tablet, notebook ou a Urna Eletrônica brasileira;
4. Mesa Receptora Independente, dispositivo que realiza a impressão, verificação, coleta ou fragmentação do voto em papel;
5. Máquina apuradora de votos para escrutínio e auditoria das cédulas de segurança.

2.1 Identificadores Biométricos do Eleitor

A Biometria já é testada desde 2010 pela Justiça Eleitoral brasileira. O Tribunal Superior Eleitoral vem promovendo o recadastramento biométrico do eleitorado em todo o Brasil, capturando suas características físicas, como a impressão digital dos dez dedos e a fotografia da face, possibilitando que essas informações sejam utilizadas pelos leitores biométricos instalados nos equipamentos de votação.

Figura 2 - Mesa Receptora e Leitores Biométricos

Na votação mecatrônica é desnecessária a figura do mesário, pois o modelo proposto é projetado para ser acionado pelo próprio eleitor e permitir o acesso à tela de votação somente após identificação do mesmo, realizada através de comparação dos dados coletados no recadastramento biométrico aos dados capturados por dois leitores, um no formato de mão (item II), para ler as impressões digitais de quatro dedos e outro, uma câmera que lê os traços da face (item I).

Com o único objetivo de automatizar o acesso ao sistema de votação, os equipamentos de votação são acoplados em um mesmo conjunto físico, cujo software de identificação biométrica, comunica o início e o término da operação ao software de votação, trocando apenas informações simples entre si, sem associar o eleitor ao voto.

Além disso, os sistemas de informática responsáveis pelo registro do voto e pela identificação do eleitor são programados separadamente e em código aberto (*Software* Livre), com total possibilidade de auditoria por equipes especializadas, reduzindo a possibilidade de fraudes, falhas ou registros que quebre o direito constitucional ao sigilo do voto e o associe ao eleitor.

A câmera digital do equipamento de votação (item I) será também utilizada com a finalidade de garantir a segurança do sigilo do voto e dificultar a retirada da cédula do ambiente de votação. Dessa forma, após realizar a identificação da face do eleitor, a câmera altera seu estado de funcionamento de um simples equipamento coletor de imagem e passa a funcionar como um sensor de presença.

O sensor de presença será responsável por garantir o sigilo da tela onde o eleitor lançará os dados dos seus candidatos, evitando que o programa de coleta de votos funcione quando for detectada a presença de mais de uma pessoa na cabina de votação.

O sensor evitará também que o eleitor deixe a cabine de votação após a impressão da cédula, ou seja, se o sistema perceber que o eleitor afastou-se da cabina de votação após a impressão da cédula, a mesa independente emitirá um sinal sonoro de alerta para que os fiscais das eleições verifiquem o motivo que levou o eleitor a interromper o procedimento do voto, garantindo o sigilo e evitando que o voto impresso saia do ambiente de votação.

2.2 Cédula de Segurança do Voto Independente

A cédula de Segurança é um papel oficial e padronizado medindo 22x9cm sensível à impressão térmica, especialmente desenvolvida para o eleitor manifestar a sua intenção de voto em eleições e/ou plebiscitos.

No Brasil, assim como as cédulas de dinheiro, ela pode conter elementos de segurança propostas pela Casa da Moeda que dificultem a adulteração ou reprodução da mesma.

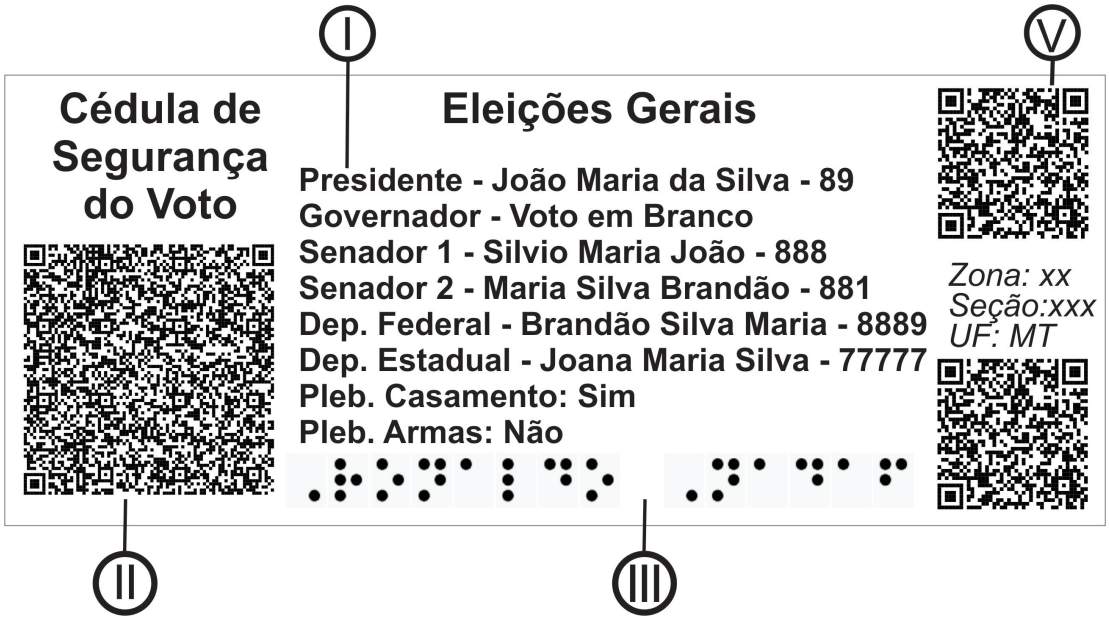


Figura 3 - Cédula de Segurança do Voto Mecatrônico

A cédula de segurança tem a característica de abrigar três registros impressos, sendo um em texto, outro em código de barras bidimensional e outro em código Braile.

Enquanto a norma norte-americana para o voto eletrônico naquele país sugere a necessidade de se gerar no mínimo dois registros [5], o modelo aqui proposto produz quatro registros do voto do eleitor, gerados por processos digitais distintos e auditáveis:

I - Um registro textual contendo o cargo e o respectivo nome e o número do candidato, gerado pela urna eletrônica após a confirmação do eleitor;

II - O segundo em código de barras bidimensional, invólucro gráfico com a dimensão de (5x5cm), localizado no canto direito da cédula, que armazena todas as informações do voto e de segurança do registro em formato XML, protegido

por assinaturas digitais de criptografia assimétricas geradas por chaves privadas institucionais instaladas nas "Mesas Receptoras Independentes".

III - Um terceiro registro na linguagem Braile, gerado por outro sistema, impresso com tinta especial em alto relevo ou impressora Braile, que garante a acessibilidade ao eleitor deficiente visual,

IV - E o quarto registro utilizado atualmente, o totalmente digital, gerado pela urna eletrônica na conclusão da votação.

V – Registros de assinaturas digitais do voto em texto gerado tanto pela entidade administradora do processo eleitoral quanto por outras entidades fiscalizadoras (3x3cm).

Nesse modelo, o voto codificado em barras bidimensional pode ser auditado pelo eleitor apenas aproximando o maior código de barras impresso na Cédula de Segurança à câmera embutida em um *tablet*, que terão softwares que vão decodificar o código bidimensional e verificar a assinatura digital institucional através da chave pública de um ou mais participantes do processo de segurança criptográfica do voto, garantindo que, além do texto com os nomes dos candidatos, outro registro no papel comprove a integridade do registro de intenção de voto.

Em testes práticos, foi possível perceber que o código de barras maior tem a capacidade de armazenar, além das informações do voto em formato XML, múltiplas assinaturas digitais. Já os códigos de barras menores podem armazenar a mesma quantidade de informações que o maior, o que possibilita a existência de mais registros do voto assinados, e com impressão bidimensional, gerados por processos digitais distintos.

2.3 Dispositivo de Registro Digital do Voto – DRE

As máquinas de votar utilizadas em todo mundo são computadores adaptados para receber o voto. Nesse sentido, o modelo é idealizado para utilizar qualquer computador ou *tablet* oferecido pelo mercado, compatível com os softwares livres a serem desenvolvidos por um consórcio de universidades brasileiras de ciências da computação, e que preencha os requisitos técnicos para os procedimentos de identificação biométrica do eleitor, impressão e validação de cédulas de segurança do voto.

Figura 4 - À esquerda, Mesa Receptora com DRE Tablet e à direita, DRE Urna Eletrônica

Outro detalhe importante é a inclusão de tela sensível ao toque, que possibilita a utilização de um software que permite ao eleitor analfabeto posicionar o dedo indicador sobre a foto do candidato e confirmar o voto. Além do teclado reaproveitado da urna eletrônica, já adaptado com códigos Braile, mantendo a mesma sistemática de votação para os eleitores deficientes visuais.

Cabe salientar que a urna eletrônica brasileira hoje em uso possui interfaces USB que podem ser conectadas a mesa receptora, e adaptada para uma fase de teste em um processo de transição para o modelo aqui idealizado [7].

2.4 Mesa Receptora Independente

A "Mesa Receptora Independente" é um equipamento com três interfaces de interação que se comunicam via USB com os sistemas eleitorais instalados nos computadores que coletam o voto digital (DRE), transmitindo comandos eletrônicos quando concluídas as tarefas de impressão, coleta ou fragmentação do voto impresso. Ela ainda acopla os leitores biométricos, a urna independente e a urna ecológica.

Figura 5 - Interfaces da Mesa Receptora

I - Interface Impressora: É de onde sai a cédula de segurança já cortada, o "Comprovante de Comparecimento", a "Zerésima" e o "Boletim de Urna".

II - Interface Coletora: Confirma a procedência através das assinaturas digitais inseridas nos códigos de barras da cédula, que é "sugada" pelo dispositivo mecânico e armazenada aleatoriamente na "Urna Independente". Concluída a operação, é emitido um sinal de autorização para a gravação e assinatura do voto digital lançado parcialmente no computador que coletou o voto.

Essa tecnologia é semelhante à utilizada nos terminais de cancelas eletrônicas para controle de bilhetes na saída de estacionamentos, que identificam o código de barras impresso do ticket entregue na entrada, comprovam a quitação, recolhem o bilhete de papel e liberam a cancela para saída.

III – Interface fragmentadora: Verifica a assinatura digital inclusa no código de barras da mesma forma que a interface coletora, confirmando a procedência, a cédula é sugada e triturada por dispositivo mecânico, e lançada em uma gaveta de armazenagem de resíduos de papel conectada a mesa receptora denominada "Urna

Ecológica", evitando que o voto rejeitado venha contaminar o processo de escrutínio ou de auditoria da eleição. Concluída a operação, um sinal é enviado para que o software no computador que coletou o voto reinicie o processo de votação, cancelando o voto digital lançado parcialmente.

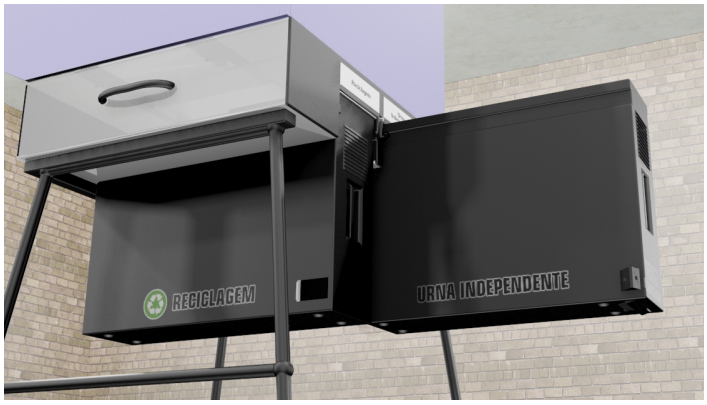


Figura 6 - Urna Independente retirada da Mesa Receptora

Para ampliar a segurança física das cédulas coletadas e armazenadas, existe uma mecânica que automatiza o lacre com tecnologia de radio frequência (RFID) da caixa coletora, acionado no final da votação, quando a "Urna Independente" será então sacada da mesa de votação e transportada para o ambiente de apuração e totalização dos votos, evitando o manuseio humano e mantendo a urna auditada contra tentativas de violação.

Importante ressaltar que, com a impressão do comprovante de comparecimento do eleitor pela própria máquina de votação, eliminam-se também os gastos atuais com a impressão e verificação dos cadernos de votação utilizados para o controle de acesso do eleitor à urna.

2.5 Máquina Apuradora

A máquina de escrutínio e auditoria é projetada para receber as cédulas de segurança resultantes da votação e armazenadas na "Urna Independente". Sua primeira função é verificar a inviolabilidade do lacre rádio frequência da caixa coletora. Confirmada a auditoria física da armazenagem, os votos são recolhidos e encaminhados para o dispositivo de leitor ótico do voto, que captura as informações do registro em código de barras e as comparam com o voto em texto lido através de tecnologia OCR.

Se o lacre da urna apresentar indícios de violação, o equipamento emite um aviso sonoro e em texto no monitor e a urna é submetida à comissão apuradora, que decide sobre o prosseguimento de sua apuração.

Figura 7 – À esquerda, Máquina Apuradora, Nos detalhes, dispositivos embaladores e cédulas lacradas

Aprovado na auditoria, o voto da cédula é então contabilizado eletronicamente e esta é enviada para um espaço onde, junto com as demais cédulas, são armazenadas e, concluída a apuração total dos votos daquela seção eleitoral, serão lacradas com plástico formando um bloco de papel que recebe uma etiqueta de rádio frequência, concluindo assim o processo de apuração eletrônica das cédulas, produzindo como resultado relatórios de apuração da seção, impressos em texto e código de barras, ou relatórios de erros na auditoria das cédulas.

Já as cédulas provenientes da auditoria manual são apuradas em outra interface da máquina. O procedimento de escrutínio é sempre iniciado após identificada a relação entre a seção eleitoral e a etiqueta RFID, que resultam relatórios e um bloco de Cédulas de Segurança com um novo identificador RFID, ou seja, um ciclo de auditoria e contagem eletrônica que pode ser controlado por relatórios que vinculam a operação de escrutínio a uma identificação RFID.

Figura 8 - Cédulas Lacradas e Controladas por Tecnologia RFID

Para efeito de segurança, a máquina não possui nenhum dispositivo de entrada, além do disco que armazena os softwares livres que controlam os equipamentos de

interpretação, auditoria e contabilidade dos votos impressos, evitando assim qualquer interferência humana ao processo de auditoria.

O modelo deve ser amparado por leis que protejam o novo registro físico do voto. O ideal é que o acesso humano às cédulas de segurança lacradas, e a contagem manual cédula por cédula, só seja permitida após algumas recontagens eletrônicas, realizadas por máquinas apuradoras diferentes ou após a constatação de inconsistências nos registros do voto impresso e/ou decisão judicial.

Nesse sentido, evita-se uma nova forma de "voto de cabresto pós-moderno", onde a correlação entre os candidatos escolhidos e expostos nas cédulas impressas serve como instrumento de coação utilizado por candidatos com má fé, pressionando eleitores com alegação de que este voto impresso poderá ser rastreado através de uma auditoria manual das cédulas de segurança.

Comprovada a necessidade através de decisão judicial de uma auditoria manual das cédulas impressas de uma seção, o primeiro escrutínio dos votos impressos será realizado por uma junta apuradora formada por deficientes visuais, que farão a leitura dos códigos Braile impressos na cédula, lançando-os em um sistema de tabulação próprio, chegando-se a totalização do resultado daquela seção.

Persistindo os pedidos para acesso às cédulas, uma segunda apuração manual, através de leitura do texto e/ou do código em barras, poderá ser realizada pelo autor do pedido sob a fiscalização do adversário, sendo obrigatório um terceiro escrutínio manual, a ser feito pelo adversário, sob a fiscalização do autor do pedido de auditoria manual dos votos.

3. Roteiro de Votação

O modelo de votação aqui concebido deve garantir a acessibilidade de todos os eleitores, inclusive os deficientes físicos, visuais e analfabetos. No caso dos deficientes físicos, anões ou pessoas mais baixas, a mesa é adaptada à altura do eleitor através de alavancas simples. No caso de deficientes visuais, os teclados que podem ser reaproveitados das urnas eletrônicas possuem linguagem Braile e a mesa uma impressora braile, tornando todos os métodos de lançamento e verificação do voto acessível.

Já no caso de eleitores analfabetos é possível visualizar a foto do candidato e escolher através da tela sensível ao toque, o que não altera é a possibilidade dele ler o nome do candidato. Vale ressaltar que todas as máquinas possirão fones de ouvidos para possibilitar que qualquer eleitor ouça o voto, além de um leitor ótico na interface trituradora que reconhece os santinhos, que também ganham um código em barras bidimensional, lendo-os digitalmente e depois dando destino ecológico através da trituração.

No processo de votação observa-se a seguinte rotina, que pode ser orientada ao eleitor através de um fone de ouvido:

a) A abertura da votação é autorizada por chaves criptográficas em código de barras bidimensional, inseridas na Mesa de Receptora pelos fiscais designados pela entidade administradora.
b) O eleitor posiciona a mão no dispositivo de identificação biométrica e faz o reconhecimento das impressões digitais. A Mesa Receptora envia uma solicitação de autorização de voto assinada digitalmente para o sistema eletrônico de votação.
c) O eleitor se posiciona em frente à câmera do computador e realiza o processo de reconhecimento biométrico da face.
d) Em caso de falso-negativo, quando não há reconhecimento biométrico do eleitor, este deve solicitar a autorização aos fiscais, que verificam a identidade e autorizam a votação. Então é feita uma fotografia da face do eleitor, a coleta das impressões digitais dos dez dedos. Em seguida é liberada a tela de votação.
e) Concluída a dupla identificação biométrica, o eleitor tem acesso à tela de votação onde visualiza os candidatos e utilizando um teclado para realizar o voto eletrônico parcial. Após confirmar os candidatos o eleitor autoriza a impressão da cédula de segurança.
f) O eleitor retira a cédula de segurança da interface impressora e audita o voto em texto e em código de barras, através de um tablet externo.
g) Confirmado o voto impresso, o eleitor insere a cédula de segurança na interface coletora, onde esta é sugada e posicionada aleatoriamente dentro da Urna Independente. Nesse momento o sistema eletrônico registra o voto digital, emitido um sinal sonoro e encerrando a possiblidade de uma nova votação para aquele eleitor.
h) Caso o eleitor se arrependa e rejeite o voto impresso na cédula, ele deve eliminá-la inserindo-a na interface fragmentadora, que tritura o papel e automaticamente emite um sinal para o programa de computador cancelar a gravação do voto digital e reiniciar o processo de votação.

Todas e quaisquer operações que não comprometam o sigilo do voto serão registradas nos arquivos de "logs" do equipamento eletrônico de votação e da Mesa Receptora Independente, sempre assinado digitalmente com as chaves privadas das instituições participantes do processo.

4. Métodos de Segurança

A segurança do modelo proposto é baseada na técnica de criptografia de algoritmos de par de chaves assimétricas onde uma informação codificada utilizando a chave privada é decodificada somente por quem possua a chave pública, e vice-versa, garantindo a integridade, procedência e o sigilo das informações geradas, quando assim for necessário.

Na cédula, ela será usada para gerar as assinaturas digitais das informações do voto impresso em texto e em linguagem XML, inclusa no código bidimensional. No caso, uma primeira assinatura que comprovará a procedência institucional será criada com a chave privada da "Mesa Independente" e uma segunda assinatura, que reforça a primeira, sendo proveniente da chave privada de uma instituição auditora do processo.

Na prática, a segurança com criptografia assimétrica funcionará da seguinte forma:

1- Cada Mesa Independente possuirá dispositivos específicos (Smart Cards, Tokens ou outro chip criptográfico) responsável pela geração do par de chaves criptográficas.
2- O programa de computador responsável pela coleta de votos, após coletar todas as intenções do eleitor para os cargos em disputa e obter a confirmação para impressão da cédula, assinará com a chave privada da Mesa Independente o registro do voto em formato de texto e o registro do voto em formato XML. Todas as assinaturas serão impressas em formato de códigos de barras.
3- Uma segunda assinatura será gerada pelo programa de coleta de votos, mas com a chave privada de uma instituição independente do administrador eleitoral.
4- Após ler o texto da cédula e confirmar seu voto, o eleitor insere a cédula de segurança na interface validadora/coletora. Um programa de computador específico, em posse da chave pública, fará a conferência da assinatura digital embutida no código de barras, confirmando que a cédula foi gerada pelo mesmo equipamento que a coletará, bem como validando a integridade das informações contidas na cédula.
5- A assinatura digital gerada pela chave privada da instituição independente poderá ser utilizada pelo eleitor para uma segunda verificação de autenticidade do voto, que será realizada por um segundo equipamento eletrônico, onde estarão armazenadas as chaves públicas.

A conversão das informações em código bidimensional QR (Quick Response) também garante a segurança do processo já que impede o eleitor de memorizar informações, como as assinaturas digitais, que identificam a cédula e poderiam ser utilizadas como comprovação da venda de votos.

Para a segurança física das cédulas coletadas serão utilizados selos, ou lacres, com tecnologia radio frequência (RFID). Esse recurso permite identificar as urnas que receberão as cédulas válidas, construindo uma correlação para auditoria entre a caixa, a seção eleitoral, a máquina que apurou as cédulas e o bloco de cédulas apuradas.

A tecnologia serve também para garantir da integridade física da urna coletora contra eventuais tentativas de violação para inclusão ou exclusão fraudulentas de cédulas.

O RFID também será utilizado na Máquina de Apuração. O equipamento que faz a auditoria e a contagem dos votos coletados pela Mesa Independente só será acionado após a comprovação da identidade da urna coletora e de sua integridade física.

Após desempenhar as funções de auditar e contar votos, a Máquina Apuradora embala as cédulas e volta a utilizar um selo RFID que, além de garantir a segurança física do escrutínio, vai ser utilizado para facilitar o arquivamento do material coletado em cada eleição.

5.Conclusão

O modelo aqui proposto vai ao encontro do projeto inicial da urna eletrônica atualmente em uso no Brasil. Quando da sua concepção, ainda na década de 80, os estudos desenvolvidos pela autoridade eleitoral avaliaram vários modelos de urnas, sendo que todos os protótipos escolhidos como matrizes do projeto continham um dispositivo de impressão para garantir a integridade da eleição em caso de falha eletrônica [7].

Retirado duas vezes do processo eleitoral brasileiro (a primeira motivada pela hoje questionada credibilidade da urna e a segunda motivada por razões técnicas e políticas), o voto impresso proposto neste artigo ressurge movido pelas mesmas preocupações dos pioneiros que alavancaram o voto eletrônico no Brasil: a segurança do processo eleitoral em caso de falha eletrônica e a transparência do processo eleitoral.

Já as máquinas aqui propostas são um desafio para as ciências, tanto para a área da computação, que poderá criar ou aperfeiçoar *softwares* de votação em código aberto, quanto para área de engenharia mecatrônica, que passa ter um norte para a inclusão da robótica no processo de votação.

Referências bibliográficas

[1] Presidência da República, Lei 12.034 disponível em http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/lei/l12034.htm

[2] Rivest, R. L. (2008) “On the notion of “software independence” in voting systems”, Phil. Trans. R. Soc. A, vol. 366 no. 1881 pag 3759-3767, disponível em http://rsta.royalsocietypublishing.org/content/366/1881/3759

[3] Supremo Tribunal Federal, Processo ADI/4543, disponível em http://redir.stf.jus.br/estfvisualizadorpub/jsp/consultarprocessoeletronico/ConsultarProcessoEletronico.jsf?seqobjetoincidente=4019347

[4] Agência Terra, Urna é a mais defasada, diz professor que violou sistema do TSE, disponível em http://noticias.terra.com.br/brasil/politica/eleicoes/urna-e-a-mais-defasada-diz-professor-que-violou-sistema-do-tse,dbeadf0a2566b310VgnCLD200000bbcceb0aRCRD.html

[5] United States Election Assistance Commision, Seção 7.8, Apêndice C da "*Voluntary Voting System Guidelines*" (vol. 1) disponível em: http://www.eac.gov/testing_and_certification/voluntary_voting_system_guidelines.aspx

[6] Tribunal Superior Eleitoral, Concorrência 11.179/2010, Especificação UE 2011, Anexo II, pag 9, disponível em http://www.tse.jus.br/transparencia/licitacoes-e-contratos/licitacoes/licitacoes-concluidas

[7] CAMARÃO, P. C. (1997), “O voto informatizado: Legitimidade Democrática”, Empresa das Artes, pag 76-81

O uso de um sistema de votação on-line para escolha do conselho universitário

Shirlei Aparecida de Chaves[1] **, Emerson Ribeiro de Mello**[1]

[1]Instituto Federal de Santa Catarina – SC –Brasil

{shirlei.chaves, mello}@ifsc.edu.br

Resumo. *A necessidade de uma participação ampla e democrática em pleitos realizados pela instituição de ensino multicampi exigia que problemas como o deslocamento de pessoas da comissão eleitoral e urnas para os municípios onde houvesse um campus ou polo de Educação à Distância, bem como a realização da apuração, não fossem impeditivos para a efetividade do processo. Este trabalho apresenta os requisitos que culminaram na escolha do sistema Helios, a personalização realizada para adequá-lo às necessidades da instituição, os resultados de seu uso na eleição do conselho universitário e os trabalhos futuros para que o mesmo possa ser ofertado como um serviço de tecnologia da informação à comunidade.*

Abstract. *The need for a broader and democratic participation in elections conducted by a multi-campi education institution demanded that issues like travel of election commission members and taking voting booth to several campuses, as well as tallying the results, did not become an obstacle to the process effectiveness. This paper presents the requirements to chose the Helios Voting System, the customisations performed in this system, the results from using it in an important election process. Our future work aim to deliver it as a technology information service to the community.*

1. Introdução

No Brasil, instituições federais de ensino são compostas basicamente pelos segmentos discente, docente e técnico-administrativo. O reitor é o cargo mais alto dentro da instituição, nomeado pelo Presidente da República para um mandato de 4 anos, após processo de consulta à comunidade interna. Como esta consulta é conduzida, depende do estatuto de cada instituição.

Ao reitor compete representar externamente a instituição, bem como administrar e superintender as atividades da mesma. Contudo, as instituições também definem em seus estatutos órgãos colegiados, como o Conselho Universitário (para as Universidades) ou Conselho Superior (para os Institutos). Este colegiado é um órgão máximo da instituição e possui caráter normativo, consultivo e deliberativo. Sua composição varia de instituição para instituição, porém é comum a todas que haja pelo menos um membro de cada segmento da comunidade interna (discentes, docentes, técnicos-administrativos), geralmente escolhidos por seus pares.

Recentemente essas instituições de ensino passaram por uma expansão e novos campi foram criados. Se até a década de 70 as instituições federais eram criadas com

um único campus por instituição, hoje algumas instituições surgem com vários campi, ou seja, vários endereços físicos espalhados pelo estado, ou mesmo, espalhados por alguns estados da federação.

Em uma instituição multicampi, o processo eleitoral tradicional para a escolha do reitor ou outro pleito de grande proporção, com cédulas em papel depositadas em urnas lacradas, torna-se mais complexo e custoso se comparado com instituições com único endereço. A distribuição das urnas, o deslocamento de pessoas da comissão eleitoral para cada um dos campi, o retorno das urnas e a apuração em si, são os principais pontos dificultadores para a realização do pleito.

Em 2011, o Instituto Federal de Santa Catarina - IFSC, deflagrou o processo eleitoral para escolha de seu reitor. Por possuir pólos de Educação a Distância (EaD) em outros estados e devido a eleição em questão ter tido candidato único, o pleito teve que ser realizado com cédulas em papel e urnas tradicionais, pois o Tribunal Regional Eleitoral (TRE), impõe as seguintes regras para uso da urna eletrônica brasileira: (1) solicitar ao TRE com uma antecedência mínima de 90 dias; (2) as urnas devem ser usadas exclusivamente dentro do estado; e (3) a disputa deve ter no mínimo dois candidatos. Apesar de neste caso o uso da urna eletrônica brasileira trazer alguns benefícios, principalmente para a apuração, ainda assim haveria a problemática de deslocamento de fiscais, servidores do quadro permanente da instituição, para todos os pólos no dia da eleição.

No IFSC, o mandato dos membros do Conselho Superior é de 2 anos e estes são escolhidos por seus pares através de um processo eleitoral. No início de 2012 em razão da dificuldade dos discentes organizarem o seu processo eleitoral, conforme prevê o estatuto, o Conselho Superior resolveu que a escolha se daria por meio de sorteio dos discentes inscritos. Contudo, a instituição comprometeu-se atuar para que nas próximas eleições os representantes discentes fossem escolhidos através de um processo eleitoral. Sendo assim, para 2014 a instituição precisaria conduzir um processo eleitoral que permitisse a toda sua comunidade interna eleger seus pares, o que inclui os alunos dos pólos de Educação a Distância.

Este trabalho apresenta os motivos pela instituição ter optado pelo sistema de votação *on-line* Helios [Adida 2008] para realizar a eleição dos representantes discentes, docentes e técnico-administrativos do Conselho Superior, bem como a personalização realizada para adequar com as necessidades da instituição, questões sobre a mudança de paradigma e os resultados obtidos.

O artigo está organizado da seguinte forma. Na Seção 2 é apresentado um relato das experiências de outras instituições de ensino na condução de seus processos eleitorais. A Seção 3 apresenta as premissas de uma eleição e diferentes tecnologias para a realização do pleito. A Seção 4 apresenta a arquitetura e as funcionalidades do sistema Helios. Na Seção 5 é apresentado o relato sobre como foi realizada a primeira eleição para escolha dos membros Conselho Superior fazendo uso de um sistema de votação *on-line*. Por fim, na Seção 6 são apresentadas as conclusões e os trabalhos futuros.

2. Experiências na literatura

Em 2011 a Sociedade Brasileira de Computação (SBC) decidiu por fazer uso do software livre para votação *on-line Helios Voting* [Adida 2008], para a escolha da Diretoria e do

Conselho da SBC para o biênio de 2011/2013. Na ocasião, estavam aptos a votar 1.757 sócios, sendo estes pertencentes a categoria sócio fundador ou efetivos, quites com a anuidade de 2011 até a data de trinta de março de 2011. Destes somente 783 de fato registraram seus votos, uma vez que o voto não era obrigatório [Pôrto et al. 2011].

A SBC também usou o Helios para escolha de seu estatuto em 2012 e novamente para escolha de sua diretoria e conselho para o biênio 2013/2015. Apesar do Helios ser oferecido na modalidade *Software as a Service* (SaaS), a SBC optou por fazer uma instalação local da solução. De acordo com observações feitas pelos autores do presente trabalho, não foram feitas personalizações no código de referência do Helios para adequá-lo a qualquer necessidade da SBC. Segundo [Cunha et al. 2013], a atual diretoria administrativa da SBC vislumbra comercializar o Helios como um serviço para outras entidades.

O sistema *Helios Voting* também foi usado pela Defensoria Pública da União para escolha do atual defensor público-geral federal. A interface com o usuário foi adaptada pela equipe do Laboratório de Tecnologias da Tomada de Decisão (Latitude) da Universidade de Brasília (UnB), permitindo que os 521 defensores públicos de todo o Brasil pudessem registrar suas escolhas por meio de qualquer dispositivo conectado à Internet [UNB 2013].

A Universidade Federal do Rio Grande do Norte (UFRN) fez uso do SIGEleição, um sistema desenvolvido internamente, em diversas eleições, entre estas a escolha de chefes de departamentos, do diretório central de estudantes e inclusive para consulta sindical de ajustes salariais. Segundo [UFRN 2012, PortalJH 2012], a facilidade e a possibilidade de ampliação da participação da comunidade foram os principais motivos para usar um sistema *on-line*.

Foi realizada uma pesquisa no mecanismo de busca do Google, para se verificar como as Universidades e os Institutos Federais realizaram a eleição do Conselho Universitário ou Conselho Superior nos últimos dois anos. Considerando-se os critérios citados, para facilitar a pesquisa, foram montadas duas *strings* de busca: `(("Instituto Federal") AND ("eleição"OR "eleições") AND "Conselho Superior"AND ("2013"OR "2014"))` para os Institutos Federais; e `(("Universidade Federal") AND ("eleição"OR "eleições") AND "Conselho Universitário"AND ("2013"OR "2014"))` para as Universidades Federais. Foi considerada até a terceira página de resultados.

Dos resultados da busca efetuada para os Institutos Federais, constatou-se, através dos editais de convocação, que os Institutos IFSP, IFF, IFAL, IFB, IFPE, IFPA, IFES, IFBaiano, IFAM, IF Sertão-PE, IFTM, IFGoiano e IFMS realizaram eleições com cédulas de papel. O IFTO e o IFBA fireram uso da Urna Eletrônica Brasileira, cedida pelo TRE.

No caso das Universidades, daquelas que foi possível extrair tal informação, constatou-se que sete delas (UFES, Unipampa, UFPel, UFFS, Unila, UFRR e UFV) realizaram a eleição através de cédulas de papel. Segundo [UFPA 2012], Universidade Federal do Pará (UFPA) fez uso do sistema SIGEleição. E a Universidade Federal do Rio Grande do Sul (UFGRS) fez uso de um sistema próprio de votação *on-line*.

3. Sistemas de votação eletrônica

Em 2012 a escolha dos membros docentes e técnico-administrativo do Conselho Superior (CONSUP) do IFSC foi realizada por meio de cédulas de papel e urnas de lona, porém os membros discentes foram escolhidos por meio de sorteio dos candidatos inscritos, como apresentado na Seção 1.

Para 2014 a instituição precisaria oferecer uma solução que permitisse aos discentes escolherem seus representantes no CONSUP através de uma eleição direta. Diante da dificuldade de realizar o certame da maneira convencional, com cédulas de papel e urnas de lona, foram realizados estudos para identificar um sistema de votação eletrônica que atendesse os seguintes requisitos:

- R.1 Só poderão votar os eleitores que forem considerados aptos pela comissão eleitoral;
- R.2 Cada eleitor só terá direito a um único voto por segmento que este estiver apto a votar (docente, discente e técnico-administrativo);
- R.3 A escolha do eleitor deve ser mantida em sigilo. Ninguém poderá saber em quem o eleitor votou, mesmo se este quiser revelar (p.e. apresentando um recibo de votação);
- R.4 A solução e o resultado da eleição devem ser auditáveis. A integridade dos votos deve ser garantida, ninguém poderá alterar, incluir ou remover votos;
- R.5 A solução deve ser economicamente viável, tanto para sua aquisição ou implantação, quanto para realização do pleito;
- R.6 A solução deve ser de fácil uso por eleitores e pela comissão eleitoral;
- R.7 Não permitir a realização de apurações parciais antes do término da eleição, visando assim garantir as mesmas chances para todos os candidatos e evitando a possibilidade de revelar escolhas de eleitores individuais.

Segundo [POST 2001], existem três tipos principais de sistemas de votação eletrônica:

Máquina de votar de gravação eletrônica direta do voto
: Eleitor faz uso de teclado ou monitor sensível ao toque para fazer suas escolhas e estas são registradas diretamente na máquina;

Contagem de cédulas realizada por máquina
: Eleitores marcam suas escolhas em cédulas de papel e as mesmas são digitalizadas para fazer a leitura ótica;

Sistemas on-line
: Eleitores poderão ir a um local físico para votar ou poderão fazer uso de qualquer computador conectado à Internet. As escolhas dos eleitores são transmitidas diretamente pela Internet para um sistema central da eleição.

A urna eletrônica brasileira é um tipo de máquina de votar de gravação eletrônica direta do voto (*Direct Record Electronic* – DRE). Seu uso na referida eleição teria como benefícios a facilidade de uso, agilidade na apuração e o fato que a maioria dos eleitores já estarem habituados com a mesma. Ou seja, muitos confiam na integridade deste equipamento, apesar de estudos apontarem fragilidades na solução [Kohno et al. 2004, Dill et al. 2003]. De qualquer forma, esta opção foi desconsiderada,

uma vez que o Tribunal Regional Eleitoral de Santa Catarina, em resposta ao ofício enviado pelo IFSC, indicou que as urnas só poderiam ser usadas dentro do estado, o que não permitiria levá-las para os pólos EaD fora do estado.

A contagem de cédulas realizada por máquina, como é visto no sistema de votação Scantegrity [Chaum et al. 2008], apesar de impor uma maior confiança ao eleitor, uma vez que atende o "princípio da independência do software em sistemas de votação" [Rivest 2008], não apresentou ser economicamente viável (requisito R.5) para a eleição em questão, pois seria necessário adquirir ou alugar equipamentos digitalizadores específicos e estes teriam que ser enviados para todos os campus e pólos EaD.

A possibilidade dos eleitores fazerem suas escolhas por meio de qualquer computador conectado à Internet, tornam os sistemas de votação *on-line* atrativos para a realização de eleições não-políticas [Qadah and Taha 2007]. Em buscas por soluções que atendessem o requisito R.5, chegou-se às seguintes opções:

Sistema Aberto de Eleições Eletrônicas (SAELE)
: Desenvolvido pela Universidade Federal do Rio Grande do Sul, trata-se de um software livre disponível no portal de software público brasileiro;

SIGEleição
: Desenvolvido pela Universidade Federal do Rio Grande do Norte, o SIGEleição[1] faz parte do Sistema Integrado de Gestão, também desenvolvido pela UFRN. Apesar de não estar sob uma licença de software livre, todo o código fonte é fornecido para instituições que firmarem acordo de cooperação com a UFRN;

Helios versão 3
: Segundo seu autor [Adida 2008], trata-se do primeiro sistema de votação *on-line* baseado na *web* e com auditoria aberta ao público (*End-to-End voter verifiable* – E2E), permitindo que:
  - Alice verifique que seu voto foi capturado;
  - Todos os votos capturados sejam exibidos publicamente em sua forma criptografada;
  - Qualquer um possa verificar que os votos capturados foram corretamente apurados.

Após análise dos códigos do SAELE e SIGEleição, conclui-se que ambos não fornecem garantias necessárias para atender os requisitos R.3 e R.7, e atendem parcialmente o requisito R.4. Os votos são registrados em claro no banco de dados e o resultado de uma eleição estaria passível de adulteração, sem gerar provas necessárias para uma auditoria.

Por outro lado, o Helios faz uso de mecanismos criptográficos para prover uma solução simples, baseada na *web* e que permite a qualquer um verificar a integridade de uma eleição, mesmo se a instalação do Helios estiver completamente comprometida [Adida 2008, Joaquim et al. 2013].

Com o Helios é possível carregar a lista de eleitores de uma eleição através de um arquivo CSV[2] no formato (login, e-mail, nome completo), contemplando o requisito R.1. O requisito R.2 também é satisfeito, pois é possível criar e conduzir ao mesmo tempo quantas eleições se desejar.

[1] https://www.sigeleicao.ufrn.br/sigeleicao

[2] *Comma-separated values*

A cédula de uma eleição pode ser acessada e preenchida por qualquer um que tiver o endereço da eleição. O eleitor pode fazer suas escolhas e o sistema cifra a cédula diretamente no navegador *web*, fazendo uso de rotinas em *javascript*. A autenticação do eleitor só é exigida no momento que este for depositar a cédula na urna. O Helios ou mesmo um atacante, não teria como descobrir as escolhas que o eleitor fez[3]. Neste ponto, pode-se deduzir que durante a transmissão da cédula pela rede, o requisito R.3 estaria sendo atendido, pois haveria proteção contra interceptação do tráfego entre o computador usado pelo eleitor e o Helios, tendo em vista que a cédula estaria cifrada e sabendo ainda que o Helios estaria sendo executado sobre o SSL/TLS.

Segundo [Jonker et al. 2013], a versão 3 do Helios apresentou algumas melhorias para lidar com ataques contra a privacidade das versões anteriores. A atual versão do Helios faz uso de criptografia homomórfica [Rivest et al. 1978], o que permite apurar todos os votos cifrados (conhecer o resultado da eleição), sem que nenhuma parte possa revelar as escolhas em cada voto individual. Sendo assim, depois da cédula depositada, ninguém poderia descobrir as escolhas do eleitor, mesmo que tenha acesso a base de dados do Helios. Isto atende o requisito R.3. Cabe frisar que o Helios foi projetado para eleições com baixo risco de coação dos eleitores, sendo este o cenário da eleição em questão, a qual é uma eleição não política conforme descrito em [Qadah and Taha 2007].

Segundo [Adida 2008, Joaquim et al. 2013], com o Helios o eleitor pode verificar que seu voto foi corretamente registrado; todos os votos (em sua forma cifrada) podem ser vistos por qualquer um; e qualquer um poderá verificar se todos os votos registrados foram corretamente apurados. E por ainda ser software livre e por fornecer documentação técnica[4] necessária para validação, pode-se afirmar que a solução atende o requisito R.4.

O Helios faz uso de criptografia de limiar [Shamir 1979] no processo de apuração. Assim, vários pessoas podem ser cadastradas como apuradores e todas precisam atuar em conjunto para realizar a apuração. Isto impõe um dificultador para a quebra do sigilo do voto dos eleitores, o que contempla o requisito R.7.

Os trabalhos [Karayumak et al. 2011, Weber and Hengartner 2009] avaliaram a usabilidade do Helios, sendo este ponto o menos favorável para a solução. As críticas estão voltadas para a interface que apresenta termos técnicos ligados a criptografia e o uso de rotinas em *javascript*, que dependendo do computador e do navegador *web* do eleitor, podem dificultar o processo para depositar a cédula na urna.

De todos os requisitos apresentados nesta seção (R.1 a R.7), pode-se afirmar que o Helios não atende plenamente o R.6 e precisaria de melhorias em sua interface com o usuário, seja este um eleitor ou o administrador de uma eleição. Desta forma, optou-se pelo Helios como o sistema de votação *on-line* a ser usado na eleição para escolha dos membros do Conselho Superior no IFSC. A Seção 4 apresenta o trabalho desenvolvido para melhorar a usabilidade do sistema, além de outras personalizações.

[3]Assumindo que não exista qualquer *software* malicioso hospedado e em execução no computador do eleitor.

[4]http://documentation.heliosvoting.org/verification-specs/helios-v3-verification-specs

4. Personalização e melhorias no sistema de votação Helios

O código fonte do Helios está disponível no Github e pode-se observar que seu desenvolvimento continua ativo[5]. Para o desenvolvimento de qualquer melhoria no Helios deve-se levar em consideração este fato, pois caso contrário as melhorias realizadas poderiam inviabilizar a atualização para futuras versões do projeto original do Helios.

Diante do exposto foi criada uma ramificação[6] (*fork*) do projeto original. Como o Git permite que se informe qual é o projeto base desta ramificação, então é possível fazer um realinhamento com o código do projeto base sempre que for desejado, por exemplo, quando são lançadas novas versões do Helios. Durante o desenvolvimento deste trabalho foi necessário realizar dois realinhamentos, os quais resultaram em alguns poucos conflitos de código, mas que puderam ser corrigidos facilmente.

A interface *web* do Helios, vista por eleitores e administradores de eleições, foi desenvolvida na linguagem Python, fazendo uso do *framework* Django, e combinada com rotinas em Javascript, estas usadas para garantir que processos de criptografia sejam realizadas diretamente no navegador do usuário e não no servidor onde o Helios está hospedado. A arquitetura do Helios é composta por quatro grande componentes [Adida 2008]:

Administrador da Eleição – Página *web* usada por usuários para criarem e gerenciarem suas eleições, informando as questões que farão parte da eleição, a lista de eleitores, a lista de apuradores, envio de e-mail para eleitores e apuradores, etc;

Cabine de Votação – Página *web* pra que os eleitores possam fazer suas escolhas, se autenticarem e por fim, depositarem sua cédula na urna, ou seja, submeter suas escolhas para o servidor onde o Helios está hospedado;

Servidor de depósito de cédulas – Responsável por receber e computar as cédulas recebidas. Trata-se de um processo executado no servidor onde o Helios está hospedado;

Centro de auditoria – Página *web* que permite a qualquer usuário auditar todas as partes da eleição. Se uma eleição já estiver encerrada, o centro permite ao usuário baixar todas as cédulas daquela eleição e realizar localmente a apuração.

As subseções a seguir apresentam os motivos e as melhorias que foram desenvolvidas nestes componentes.

4.1. Tradução de interface e usabilidade

O *framework* Django fornece suporte à internacionalização[7], contudo os componentes *web* do Helios não foram codificados para usufruir desta funcionalidade e, por consequência, não oferecem qualquer facilidade que permita traduzir as mensagens das interfaces com o usuário. Outros componentes se quer possuem suporte à internacionalização, como é o caso da Cabine de Votação, cujos os respectivos arquivos HTML são servidos estaticamente pelo servidor *web* e não passam pelo processador de modelos do Django.

Embora seja possível fazer a marcação de todas as *strings* que aparecem na interface do usuário (eleitor ou administrador), para depois relacioná-las com arquivos

[5]https://github.com/benadida/helios-server/graphs/contributors

[6]https://github.com/shirlei/helios-server

[7]Adaptação das mensagens de um sistema para diferentes línguas.

de tradução do Django, ainda assim não atenderia todas as partes do sistema, p.e. Cabine de Votação. Sendo assim, optou-se neste primeiro momento por uma solução mista. Onde era possível utilizar as facilidades do Django para tradução, o mecanismo de internacionalização do mesmo foi utilizado (maior parte do sistema). Onde não era possível (Cabine de Votação), a tradução foi feita diretamente nos documentos HTML e nas rotinas Javascript dos mesmos.

Conforme apresentado na Seção 3, a usabilidade é o ponto menos favorável do Helios. Para contornar essa questão, uma das medidas tomadas foi a de, durante a tradução, adaptar o texto para ficar mais amigável ao usuário, pois o texto original tinha como principal objetivo apresentar uma explicação sobre como os requisitos de segurança estão sendo garantidos pelo sistema.

A interface de gerenciamento de eleições foi reorganizada, de modo a facilitar a visualização de ações a serem tomadas. Também iniciou-se processo de utilização do Bootstrap[8] como *framework* de desenvolvimento da interface. Este fornece diversas facilidades para tratamento de estilos de elementos comuns como formulários e botões, além de *plugins* Javascript para menus e ainda, é preparado para operar com *layouts* responsivos, ou seja, que adequam a apresentação da página *web* de acordo com o tamanho da tela do dispositivo do usuário.

4.2. Módulo super administrador

Na forma como o Helios é oferecido em seu sítio *web*, qualquer pessoa pode usar o sistema para criar e gerenciar suas próprias eleições, para isto basta que possua uma conta de usuário em um dos serviços: Google, Facebook ou Yahoo. No caso tratado por este artigo, foi realizada uma instalação local do sistema e tinha-se como necessidade garantir que somente usuários autorizados previamente pudessem criar e gerenciar suas eleições.

O *framework* Django trabalha com o conceito de aplicativos Django (*django apps*), ou seja, componentes de software com regras bem definidas e auto contidos. A instalação padrão do Django possui diversos aplicativos, entre estes o *admin app*. Este aplicativo fornece uma interface automática de administração a qual permite ler metadados dos modelos de projeto, bem como inserir conteúdo nestes modelos. Por padrão este módulo está desabilitado no Helios.

Com a habilitação deste módulo e com algumas modificações, foi possível criar uma área chamada de "super administração". Voltado para a equipe de administração do serviço Helios, este módulo oferece funcionalidades para inserir, alterar e remover usuários da lista de autorizados a gerenciar eleições no sistema.

Como informado na Seção 3, o Helios é um sistema de eleição *on-line* totalmente verificável, porém a auditoria se resume nas informações da própria eleição e não abrange questões, como por exemplo, sobre como e quando o sistema foi acessado pelo administrador de uma eleição.

No portal do super administrador foi criada uma área de auditoria para registrar as ações tomadas pelo administrador de uma eleição. O código padrão do Helios permite ao administrador de uma eleição, enquanto esta estiver aberta, adicionar ou remover eleitores da lista de eleitores aptos a votar. Com o centro de auditoria desenvolvido neste trabalho,

[8] http://getbootstrap.com

toda ação administrativa de remoção de eleitor é registrada, e cada registro contém dados sobre qual eleitor foi removido, data e hora da ação e qual administrador fez tal ação.

Nas modificações realizadas incluiu-se também características relativas ao registro dos votos depositados pelos eleitores. No caso, o Helios registrava, além do voto, a data e a hora do mesmo. Adicionalmente passou-se a registrar também o endereço IP do computador usado pelo eleitor. Estas informações foram disponibilizadas na interface do super administrador.

Cabe frisar que tais registros não ferem a privacidade dos eleitores, pois não é possível através destes determinar em quem o eleitor votou. O objetivo desta funcionalidade é para gerar uma ferramenta de investigação que possa ser usada diante de uma possível disputa. Por exemplo, verificar se um mesmo computador foi usado para registrar vários votos em um curto espaço de tempo, podendo assim caracterizar coação de um conjunto de eleitores.

4.3. Autenticação de usuários no serviço de diretórios LDAP

Primeiramente, é importante destacar que o Helios não permite o voto anônimo, ou seja, todo eleitor deve passar por um processo de autenticação para provar sua identidade. Ao criar uma eleição é possível indicar se a mesma é uma eleição fechada, isto é, só poderão votar os eleitores que foram carregados através de um arquivo CSV, ou se é uma eleição aberta, isto é, poderá votar qualquer pessoa que consiga se autenticar através dos mecanismos que o Helios provê suporte.

Como optou-se por fazer uma instalação local do Helios e os administradores de eleições deveriam ser obrigatoriamente servidores públicos da instituição, a opção de autenticar através de serviços como Google, Facebook e Yahoo foi considerada inadequada. O mesmo para as eleições públicas, pois é desejado que somente pessoas que possuam credenciais da instituição possam votar.

A instituição possui um serviço de diretórios LDAP [Wahl et al. 1997] o qual é usado por todos os sistemas de informação. Sendo assim, o Helios precisaria também autenticar seus usuários através do LDAP.

O Helios provê um módulo de autenticação que inicialmente só permite autenticação através de nome de usuário e senha, armazenados em uma base local, e através do OAuth [Hammer-Lahav 2010], usado para Google, Facebook e Yahoo. Este módulo foi então estendido para permitir a autenticação de usuários, eleitores ou administradores, presentes em uma base LDAP.

5. Condução e resultados da eleição com sistema de votação *on-line*

A mudança de paradigma é algo que pode gerar resistência por parte dos envolvidos. Visando minimizar uma possível resistência ou mesmo questionamento sobre a lisura do processo, fora apresentado para o Comitê Gestor de Tecnologia da Informação e para a Comissão Eleitoral, o funcionamento do sistema, suas características e também uma analogia com a votação por meio de cédulas de papel, sobre as fases de uma eleição, que são: identificação dos eleitores, votação e apuração.

O Conselho Superior possui representantes dos segmentos discente, docente e técnico administrativos em educação, sendo estes eleitos por seus pares. Optou-se por

criar uma eleição para cada segmento, carregando para cada um arquivo CSV com a lista de eleitores aptos a votar. Estas listas foram geradas a partir de extração dos sistemas de informação da instituição e tornadas públicas no sítio *web* (*hot site*) da eleição, de forma que os interessados pudessem entrar em contato com a comissão e sugerir correções.

O trabalho de criação das eleições foi conduzido pela equipe de TI em conjunto com os membros da comissão eleitoral. O Helios permite determinar um número mínimo e máximo de respostas para cada questão. Ao invés de se considerar a opção de resposta mínima igual a zero, para englobar brancos e nulos, a comissão optou por explicitamente criar uma opção de resposta BRANCO e uma opção NULO. Desta forma, as questões foram configuradas para que o eleitor escolhesse no mínimo uma e no máximo uma resposta.

Para todas as eleições foi feito uso de pseudônimos para os eleitores e foram adicionados três apuradores, estes escolhidos entre os membros da comissão eleitoral. O par de chaves criptográficas de cada apurador foi gerado naquela ocasião, sendo que a chave privada de cada apurador foi salva em um pendrive diferente e entregue ao apurador em questão.

Ciente que seria necessário ter a chave dos três apuradores para abrir a urna e apurar os votos, uma cópia da chave privada de cada apurador também foi salva em um pendrive backup, o qual foi colocado dentro de um envelope que depois foi lacrado e assinado pelos membros da comissão. Este procedimento foi adotado como medida de segurança, caso um dos apuradores viesse a perder seu pendrive.

Nas eleições anteriores, com urna de lona, era dedicado um único dia para realizar a votação. Por esta solução depender da infraestrutura de TI onde está hospedado o Helios, optou-se por estender para 4 dias o período para votação. Assim, mesmo que houvesse uma indisponibilidade temporária do Helios ou mesmo perda de conectividade de algum campus, haveria tempo hábil para que todos eleitores pudessem depositar seus votos.

No dia e hora de abertura do período de votação, a comissão eleitoral, através do Helios, fez o envio do usuário e senha para o e-mail de cada eleitor cadastrado. Isto resultou no envio de mais de 14.347 e-mails e foram necessárias quase quatro horas para que todos os e-mails fossem entregues. Verificou-se que o motivo desta demora estava relacionado com o mecanismo de fila de tarefas distribuída[9], pois existia somente um processo responsável por todo o envio de e-mail. A solução foi iniciar cinco processos (*workers*) de forma concorrente, porém isto só foi útil para o envio de e-mails de lembrete de votação para aqueles eleitores que ainda não haviam depositado seus votos.

O processo de apuração ocorreu sem problemas, sendo feita em uma sessão pública. Na eleição para TAEs, 63% dos eleitores compareceram às urnas (568), 59% dos docentes (572) e apenas 5% dos discentes (689). Números bem próximos com os da última eleição em 2011 realizada com cédulas de papel, 65% de TAEs (451) e 64% de docentes (513). Discentes não participaram da eleição de 2011.

6. Conclusões

A principal facilidade de um sistema de votação *on-line* é a possibilidade do eleitor votar por meio de qualquer dispositivo conectado à Internet. Porém, em eleições com grande

[9]https://celery.readthedocs.org/en/latest/

probabilidade de coação de eleitores, tal solução pode ser questionada pelos interessados, principalmente sobre a garantia da pessoalidade do voto, ou seja, nenhum eleitor poderia se passar por outro.

Para este caso, poderia-se fazer uso de mecanismos de forma que o acesso ao servidor do Helios só fosse feito através de computadores previamente registrados, por meio de firewall, VPNs, etc. Desta forma, cada campus teria um computador atuando exclusivamente como urna em uma sala com controle de acesso restrito. Ou seja, para ingressar na sala o eleitor teria que assinar uma lista de presença e somente um eleitor por vez. Este computador garantiria que o eleitor só pudesse depositar o voto uma única vez.

Como trabalhos futuros pretende-se disponibilizar a solução como um serviço de TIC, exigindo um mínimo ou nenhuma interação da área de TI na configuração de cada eleição. Para isto será necessário desenvolver um mecanismo para gerar automaticamente a lista de eleitores, de acordo com a escolha do administrador da eleição. Outra melhoria importante é a de adicionar na interface de administração a possibilidade de se configurar data e hora inicial e final da eleição, permitindo que a mesma seja aberta e fechada sem a necessidade intervenção manual por parte do administrador.

Todas as alterações e personalizações descritas neste artigo foram disponibilizadas publicamente no GitHub[10] sob a mesma licença de software livre usada pelo Helios. Também foi gerado um arquivo de instruções que detalha todos os passos necessários para uma instalação e configuração funcional do sistema.

Referências

Adida, B. (2008). Helios: Web based open audit voting. In *17th USENIX Security Symposium*.

Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., and Vora, P. (2008). Scantegrity: End-to-end voter-verifiable optical-scan voting. *Security & Privacy, IEEE*, 6(3):40–46.

Cunha, P. R. F., Granville, L. Z., and de Matos Galante, R. (2013). Plano de gestão para a SBC biênio 2013-2015.

Dill, D., Mercuri, R., Neumann, P., and Wallach, D. (2003). Frequently asked questions about dre voting systems. *Verified Voting*.

Hammer-Lahav, E. (2010). The oauth 1.0 protocol.

Joaquim, R., Ferreira, P., and Ribeiro, C. (2013). Eviv: An end-to-end verifiable internet voting system. *computers & security*, 32:170–191.

Jonker, H., Mauw, S., and Pang, J. (2013). Privacy and verifiability in voting systems: Methods, developments and trends. *Computer Science Review*, 10:1–30.

Karayumak, F., Olembo, M. M., Kauer, M., and Volkamer, M. (2011). Usability analysis of helios-an open source verifiable remote electronic voting system. In *Proceedings of the 2011 USENIX Electronic Voting Technology Workshop/Workshop on Trustworthy Elections. USENIX*.

[10]https://github.com/shirlei/helios-server

Kohno, T., Stubblefield, A., Rubin, A. D., and Wallach, D. S. (2004). Analysis of an electronic voting system. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 27–40. IEEE.

PortalJH (2012). Maioria dos docentes da ufrn aprova proposta do governo federal. http://jornaldehoje.com.br/maioria-dos-docentes-da-ufrn-aprova-proposta-do-governo-federal. Visitado em agosto de 2014.

POST, U. (2001). Online voting. In *POSTNOTE*, number 155. UK Parliamentary Office of Science and Technology.

Pôrto, I. J., Galante, R., and Zorzo, A. (2011). Ata da sessão pública de apuração dos votos para eleição do conselho e da diretoria da sociedade brasileira de computação.

Qadah, G. Z. and Taha, R. (2007). Electronic voting systems: Requirements, design, and implementation. *Computer Standards & Interfaces*, 29(3):376–386.

Rivest, R. L. (2008). On the notion of 'software independence'in voting systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3759–3767.

Rivest, R. L., Adleman, L., and Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

UFPA, I. (2012). Eleição dos servidores técnico-administrativos para os conselhos superiores. http://www.itec.ufpa.br/index.php?option=com_content&view=article&id=552:eleicao-dos-servidores-tecnico-administrativos-para-os-conselhos-superiores&catid=47:noticias&Itemid=192. Visitado em agosto de 2014.

UFRN (2012). Eleita a nova diretoria do dce com o uso do sigeleição. http://sistemasdaufrn.blogspot.com.br/2012/10/eleita-nova-diretoria-do-dce-com-o-uso.html. Visitado em agosto de 2014.

UNB (2013). Unb adapta sistema de voto eletrônico para defensoria pública da união. http://www.unbciencia.unb.br/index.php?option=com_content&view=article&id=605%3Aunb-adapta-sistema-de-voto-eletronico-para-defensoria-publica-da-uniao&catid=43%3Aeletrica&Itemid=9. Visitado em agosto de 2014.

Wahl, M., Howes, T., and Kille, S. (1997). Lightweight directory access protocol (v3).

Weber, J. and Hengartner, U. (2009). Usability study of the open audit voting system helios. www.jannaweber.com/wp-content/uploads/2009/09/858Helios.pdf. Visitado em setembro de 2014.

SBSeg 2014 — Belo Horizonte, MG

XIV Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

CTD – III Concurso de Teses e Dissertações Computacional

Segurança em Redes-em-Chip: Mecanismos para Proteger a Rede SoCIN contra Ataques de Negação de Serviço[1]

Sidnei Baron, Michelle Silva Wangham, Cesar Albenes Zeferino
Laboratório de Sistemas Embarcados e Distribuídos
Universidade do Vale do Itajaí
Itajaí – SC – Brazil
{sbaron,wangham,zeferino}@univali.br

Abstract. *Networks-on-Chip (NoCs) are vulnerable to security attacks. In this context, this work aimed at increasing the availability of a NoC by means of the implementation of hardware-based mechanisms that filter malicious packets injected into the network by an attacking core. The mechanisms discard packets that affect the network availability, and regulate the injection rate of communication flows that attempt to consume a bandwidth higher than a limit specified by the system designer. The security mechanisms were described in VHDL and synthesized to an ASIC technology. Results demonstrate that they are effective in improving the network availability, with a reduced silicon overhead and low impact to the NoC performance.*

Resumo. *NoCs (Networks-on-Chip) são vulneráveis a ataques de segurança. Neste contexto, este trabalho tem como objetivo aumentar a disponibilidade de uma NoC por meio da implementação de mecanismos em hardware que filtram pacotes maliciosos injetados na rede. Os mecanismos regulam a taxa de injeção dos fluxos de comunicação que tentam consumir uma largura de banda maior do que o limite definido pelo projetista e descartam os pacotes que afetam a disponibilidade da rede. Os mecanismos de segurança foram descritos em VHDL e sintetizados na tecnologia ASIC. Os resultados demonstram a efetividade na melhoria da disponibilidade da rede, com uma sobrecarga de silício reduzida e baixo impacto para o desempenho da NoC.*

1 Introdução

Com o advento dos processos submicrônicos, a capacidade de integração de transistores tem atingido níveis que possibilitam a construção de sistemas computacionais completos em uma única pastilha de silício, os quais são denominados sistemas integrados ou SoCs (System-on-Chip) [Martin e Chang 2003].

Com o aumento do número de núcleos em um único chip, as necessidades de escalabilidade de comunicação e de largura de banda se tornaram criticamente importantes. Para suprir essas necessidades, as redes chaveadas *intrachip* estão substituindo os barramentos e canais ponto-a-ponto na comunicação entre os núcleos de um sistema integrado [Jerger e Peh 2009]. Essas redes são denominadas NoCs (Networks-on-Chip) ou Redes-em-Chip.

[1] Este trabalho foi realizado com recursos do CNPq.

Como todos os sistemas computacionais, um SoC, baseado em NoC ou não, também é alvo de ataques à sua segurança. Por exemplo, os invasores podem tentar extrair alguma informação do sistema com o objetivo de obter dados pessoais dos seus usuários ou para ignorar algumas licenças de uso de algum software. Outro possível ataque visa degradar o desempenho do sistema por meio de ações que levem à negação de serviços dentro do SoC.

Em um SoC que utilize uma NoC como infraestrutura de comunicação, a rede é o coração do sistema, pois gerencia todas as comunicações. É por isso que os ataques contra a NoC são críticos [Texier 2009]. Por meio da implementação de mecanismos de segurança nos componentes da rede, é possível bloquear ataques de uma tarefa atacante contra uma tarefa vítima [Sepulveda, Strum e Chau 2009].

Um exemplo de NoC é a SoCIN (SoC Interconnection Network), uma rede parametrizável de baixo custo para interconexão de núcleos em SoCs [Zeferino e Susin 2003]. Essa rede não possui mecanismos que garantam suas propriedades de segurança. Portanto, um sistema que a utilize está vulnerável a ataques de segurança. Dentre os diferente ataques, a SoCIN é especialmente vulnerável a ataques de negação de serviço que visem reduzir a disponibilidade do sistema.

Este artigo apresenta os resultados de uma dissertação de mestrado[2] desenvolvida com o objetivo de reduzir as vulnerabilidades da SoCIN a ataques de negação de serviço e ataque de disfarce com o uso de mecanismos de controle de acesso e formatação de tráfego. A metodologia aplicada na dissertação baseou-se na análise das vulnerabilidades da SoCIN e na seleção e implementação em hardware de mecanismos de segurança para bloquear os ataques considerados.

O restante deste artigo está organizado em seis seções. A Seção 2 apresenta definições básicas sobre NoCs e discute aspectos sobre segurança em NoCs, identificando os trabalhos relacionados. A Seção 3 discute as vulnerabilidades da SoCIN e apresenta os mecanismos propostos para aumentar a sua segurança. Na Seção 4, são fornecidos detalhes da implementação e da avaliação desses mecanismos. A Seção 5 apresenta os resultados obtidos, enquanto a Seção 6 identifica as principais contribuições da dissertação. Por fim, a Seção 7 apresenta as conclusões do trabalho.

2 Segurança em Redes-em-Chip

2.1 Definição de Rede-em-Chip

Uma NoC é composta basicamente de três componentes: interface de rede, enlace e roteador. A interface de rede, ou NI (Network Interface), implementa o protocolo que realiza a adaptação entre os protocolos de comunicação dos núcleos e o da NoC [Benini e De Micheli 2006]. Um enlace liga dois pontos na rede, sendo que esses pontos podem ser um roteador ou um núcleo. Um roteador é composto de uma estrutura de chaveamento (denominada *crossbar*) e uma lógica de controle para roteamento e arbitragem, além de portas para comunicação com outros roteadores e/ou com os núcleos (porta local) [Jerger e Peh 2009].

Uma NoC pode ser descrita pela sua topologia e pelos seus mecanismos de comunicação [Zeferino 2003]. A topologia determina a sua organização física e os

[2] http://www.univali.br/Lists/TrabalhosMestrado/Attachments/748/Sidnei%20Baron.pdf

mecanismos de comunicação definem a forma pela qual ocorre a transferência das mensagens pela rede. Os mecanismos de comunicação incluem: controle de fluxo, chaveamento, memorização, roteamento e arbitragem.

Uma mensagem transferida pela NoC é quebrada em pacotes e os pacotes, por sua vez, são divididos em unidades de transferência de tamanho fixo denominadas *flits* (*flow control units*). Um pacote é formado por *flits* de cabeçalho, corpo e cauda. O cabeçalho contém as informações necessárias ao roteamento, como, por exemplo, o endereço do nodo destinatário e o endereço do nodo origem. O corpo do pacote, também chamado de carga útil, contém os dados da mensagem. Já a cauda, também denominada terminador, marca o fim do pacote e pode conter dados da mensagem ou um código para verificação da integridade do pacote após a sua transferência.

2.2 Trabalhos relacionados

Um levantamento bibliográfico foi realizado na dissertação para caracterizar quais técnicas têm sido utilizadas para prover segurança em SoCs baseados em NoC [Baron, Wangham e Zeferino 2014]. Desse estudo, foi possível identificar que: (*i*) os principais ataques tratados pelas pesquisas atuais são os ataques de modificação de mensagem, de revelação do conteúdo da mensagem e negação de serviço; (*ii*) as propriedades de segurança mais abordadas são as de integridade, confidencialidade e disponibilidade; (*iii*) os mecanismos mais utilizados são os de integridade dos dados e de controle de acesso; e (*iv*) a interface de rede é componente mais utilizado na implementação das soluções de provimento de segurança nos SoCs, principalmente os baseados em NoCs.

No âmbito específico de NoCs, ataques de negação de serviço (Denial of Service – DoS) foram abordados por alguns autores. Diguet et al. (2007) apresentaram um mecanismo de autenticação para permitir que somente transações autorizadas sejam liberadas pela interface de rede na NoC. No trabalho de Fiorin, Silvano e Sami (2007), foram propostos mecanismos de QoS para limitar a largura de banda por núcleo, evitando o consumo excessivo dos recursos da rede. Em outro trabalho, Fiorin, Palermo e Silvano (2008) propuseram a inclusão de um mecanismo na interface de rede para detectar e bloquear comportamento não natural do tráfego destinado ao núcleo. No trabalho de Sepúlveda, Strum e Chau (2009), foi abordada a implementação de QoS para garantir largura de banda para o tráfego de dados críticos. Com abordagem semelhante, Sepúlveda, Strum e Chau (2010) adotaram uma combinação de técnicas de chaveamento por circuitos e por pacotes para distinção entre os tráfegos de dados críticos e não críticos. Já Lukovic e Christianos (2010) implementaram um mecanismo para analisar a execução da aplicação no núcleo, validando se o comportamento executado é o comportamento esperado.

A dissertação descrita neste artigo teve como objetivo desenvolver soluções de segurança contra ataques de DoS a NoC SoCIN que incluem: (*i*) controle de acesso: controle do pacote que é injetado à rede mediante a verificação prévia da origem e do destino do pacote e o consequente descarte do pacote quando este violar regras de segurança; e (*ii*) formatação de tráfego: controle da largura de banda utilizada pelos fluxos de comunicação de modo a assegurar que nenhum núcleo injete pacotes na rede a uma taxa maior do que a que foi dimensionada pelo projetista do sistema ou bloqueie recursos da rede pelo envio de pacote sem terminador.

3 Adicionando segurança à SoCIN

No projeto original da SoCIN, não foram levados em consideração aspectos específicos relacionados à segurança. Embora alguns tipos de ataques possam ser evitados pela natureza dos mecanismos de comunicação utilizados (ex. roteamento ou arbitragem), esta é vulnerável a outros tipos de ataques.

A análise das vulnerabilidades da SoCIN foi realizada com base nas características de roteamento da SoCIN. Ataques referentes à aplicação (*e.g.* acesso a uma região da memória de um núcleo) ficaram de fora do escopo da dissertação. Como resultado da análise, pôde-se observar algumas das vulnerabilidades da SoCIN, conforme descrito a seguir[3].

O envio de pacotes para um endereço de destino inválido é um tipo de ataque que pode ocorrer quando um núcleo envia um pacote para o seu próprio endereço ou para um endereço além das fronteiras da rede, bloqueando sua porta de acesso à rede ou outros recursos (*buffers* dos roteadores). Esse tipo de ataque pode ocorrer quando o núcleo ou a interface de rede possuem uma tabela de roteamento reconfigurável em tempo de execução e um código malicioso obtém acesso privilegiado para modificá-la, o que permite a definição de um endereço destino inválido.

Um ataque de repetição do envio de pacote para um determinado endereço ocorre quando um núcleo envia vários pacotes repetidamente a um mesmo endereço na tentativa de tornar o núcleo de destino e/ou a rede indisponível.

O envio de pacote sem terminador ocorre quando um núcleo injeta um pacote na rede, porém nunca envia o seu terminador. O terminador serve para liberar os *buffers* alocados nos roteadores para a transmissão do pacote. Sem o envio do terminador, o espaço em memória alocada para o recebimento dos *flits* no núcleo de destino poderá chegar ao seu limite, não permitindo o recebimento de novos pacotes. Além de indisponibilizar o núcleo de destino, os canais alocados pelo pacote não são liberados e todos os recursos do caminho entre os nodos origem e destino mantém-se bloqueados.

Já um ataque de disfarce ocorre quando um núcleo envia um pacote de requisição com um endereço origem diferente do seu próprio endereço. O pacote é encaminhado ao núcleo destino que, após receber o pacote, realiza o serviço requisitado e envia uma resposta a um terceiro núcleo que não está aguardando por essa resposta. Além de consumir largura de banda da rede, esse ataque pode levar a um mal funcionamento dos núcleos alvos do ataque.

4 Implementação e Avaliação

Nesta seção, são apresentados os dois mecanismos propostos na dissertação para aumentar a segurança da SoCIN contra ataques de negação de serviço (DoS). Os mecanismos foram integrados em um componente denominado SEW (SEcurity Wrapper), conectado entre o núcleo e o roteador (Figura 1.a). O componente é dividido em dois módulos de hardware (*wrappers*): um que realiza o controle de acesso, filtrando os pacotes maliciosos com endereços inválidos, e outro que realiza a formatação de tráfego, regulando o consumo da largura de banda pelo núcleo.

[3] Detalhes sobre os modelos são apresentados na dissertação (BARON, 2013).

A Figura 1.b apresenta a estrutura interna do SEW, o qual é composto de dois *wrappers* (nomeados de Wrapper 1 e Wrapper 2), um *buffer* do tipo FIFO (First-In First-Out) e um módulo de controle de fluxo, denominado OFC (Output Flow Controller), que regula o tráfego dos *flits*. A inclusão da FIFO e da OFC foi necessária para fazer um controle de fluxo em nível de *flit*, mas, em uma futura implementação, esse controle poderá ser integrado aos componentes já existentes da interface de rede ou da porta local do roteador. O sinal de saída IRQ (Interrupt Request Line) é utilizado para sinalizar o núcleo de que um ataque foi detectado e uma ação de segurança foi realizada, como, por exemplo o descarte do pacote.

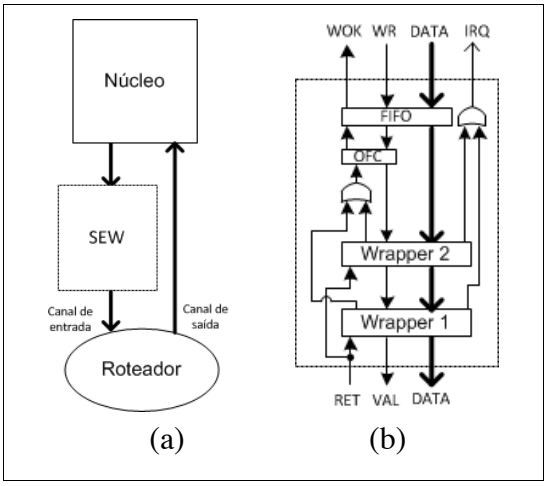


Figure 1. Implementação do mecanismo de segurança da SoCIN

O Wrapper 1 realiza o controle de acesso, filtrando pacotes com endereço de destino inválido. Se o pacote estiver endereçado para uma posição inválida (ou seja, endereço do próprio núcleo ou endereço além das fronteiras da rede), o pacote é descartado, não consumindo os recursos da rede. Neste mesmo *wrapper*, é verificado se o endereço de origem do pacote é realmente o endereço deste núcleo, ou seja, do núcleo local. Com essa implementação, qualquer tentativa de mascaramento de endereço de origem (disfarce) pode ser evitada. Somente os pacotes com as origens corretas são injetados na rede, caso contrário, são descartados.

O Wrapper 2 realiza a formatação de tráfego, controlando a banda utilizada pelos fluxos de comunicação gerados pelo núcleo e evitando ataques de envio de pacotes repetidamente com o intuito de causar congestionamentos na rede. Este *wrapper* contabiliza a quantidade de ciclos de relógio gastos para enviar um pacote ao roteador, retardando o envio do próximo pacote por um período mínimo proporcional à largura de banda máxima alocada pelo projetista do sistema ao núcleo. Este *wrapper* também monitora o tamanho de um pacote que é injetado e trunca o pacote, caso este ultrapasse o tamanho máximo predefinido pelo projetista do sistema. O *wrapper* injeta um terminador para marcar o final do pacote e todos os novos *flits* recebidos do núcleo são descartados até que o seu terminador seja recebido pelo *wrapper*, caracterizando o fim do pacote. Em um ataque, é possível que o terminador do pacote nunca seja recebido pelo *wrapper*. Neste caso, o núcleo infectado com código mal intencionado poderá ficar bloqueado indefinidamente, caso a rotina de tratamento da interrupção do *wrapper* não tome a medida necessária para evitar esse bloqueio (*i.e.* injeção de um terminador).

5 Resultados

O SEW foi implementado em VHDL e sintetizado em uma tecnologia ASIC (Application-Specific Integrated Circuit) de 90nm, utilizando-se a ferramenta Design Compiler da Synopsys e a biblioteca SAED90nm. O resultado da síntese foi comparado com o de um roteador de 5 portas da SoCIN, o qual foi configurado para o uso de canais de dados de 32 bits e *buffers* FIFO de 4 *flits*. O experimento mostrou que o módulo SEW adiciona um custo reduzido ao sistema (somente 4,1% de área de silício e 2,5% de dissipação de potência). Comparando os *wrappers* que compõe o SEW, o Wrapper 2 é o responsável pela maior parte dos custos, pois utiliza circuitos mais complexos.

A solução proposta apresentou um baixo impacto no desempenho da rede. O módulo SEW adiciona somente um ciclo de relógio de latência por pacote e não degrada a frequência máxima de operação da rede. O SEW pode operar a uma frequência de 250 MHz, enquanto a frequência máxima de operação do roteador é igual a 218 MHz.

A verificação do correto funcionamento dos circuitos implementados e a confirmação do atendimento das suas especificações foi feita com base em simulação funcional em VHDL, utilizando-se a ferramenta ModelSim.

Para verificar a efetividade das soluções propostas em um sistema, foi realizada a descrição do SEW em SystemC no nível de transferência entre registradores. Então, foi modelado um SoC composto de uma rede SoCIN 3x3, um núcleo malicioso, executando os ataques, e oito geradores de tráfego. Foram realizados experimentos com a rede sem os mecanismos de segurança e com a rede protegida pelo SEW, o qual foi utilizado para conectar o núcleo malicioso à NoC. Os experimentos demostraram a eficiência dos mecanismos propostos, os quais ofereceram uma cobertura de 100% na proteção da rede contra os ataques considerados.

6 Contribuições da dissertação

A principal contribuição da dissertação residiu na análise das vulnerabilidades de uma NoC e a proposta de soluções que a protegem contra ataques a sua propriedade de disponibilidade. Embora as soluções propostas tenham visado uma NoC específica, entende-se que podem ser aplicadas a outras NoCs com características similares.

Em relação aos demais trabalhos descritos na literatura, esta dissertação tem como diferencial o fato de evitar que os pacotes maliciosos sejam injetados na rede. Muitas das soluções existentes filtram pacotes quando eles chegam no destinatário e não quando são injetados pelo nodo origem. Assim, não evitam que os pacotes maliciosos desperdicem largura de banda da rede.

Este trabalho foi a primeira dissertação com esse foco realizada no grupo de pesquisa no qual está inserido. A partir deste, foi iniciada uma nova linha de pesquisa e, atualmente, estão em andamento outros estudos em nível de graduação e de pós-graduação. Ressalta-se ainda que, pelo nosso conhecimento, esta dissertação foi o segundo trabalho acadêmico em nível de pós-graduação sobre segurança em NoCs desenvolvido no Brasil[4].

[4] O primeiro trabalho foi uma tese de doutorado da Universidade de São Paulo [Sepúlveda 2011].

Com relação às publicações, dois artigos foram derivados da dissertação[5]. O primeiro foi publicado na IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2013) e apresenta os mecanismos propostos, detalhando a implementação dos *wrappers* de segurança [Baron, Wangham e Zeferino 2013][6]. Essa conferência é atualmente classificada no estrato B1 do Qualis de Conferências da área de Ciência da Computação[7]. O segundo artigo foi publicado na Revista de Informática Teórica e Aplicada (RITA) e apresenta uma revisão da evolução das pesquisas sobre segurança em NoCs e uma análise do estado da arte sobre esse tema [Baron, Wangham e Zeferino 2014][8]. Esse periódico é atualmente classificado no estrato B4 do Qualis de Periódicos da área de Ciência da Computação.

7 Conclusões

O objetivo deste trabalho foi buscar soluções de segurança para prover maior disponibilidade à rede SoCIN. As vulnerabilidades da SoCIN foram identificadas e soluções foram implementadas para garantir a propriedade de disponibilidade na rede contra alguns ataques de negação de serviço. As soluções propostas foram efetivas e apresentaram baixo impacto nos custos e no desempenho da rede.

Como trabalhos futuros, propõe-se a integração dos mecanismos propostos na interface de rede ou na porta local do roteador. Propõe-se também a implementação de um mecanismo pós-ataque para colocar em quarentena o núcleo malicioso e a implementação de um mecanismo de segurança dinâmico que, ao invés de utilizar um parâmetro estático de controle de banda definido em tempo de projeto, avalie se a quantidade de pacotes enviada oferece riscos à rede e regule a largura de banda alocada dinamicamente, alterando as taxas de inserção de ciclos de espera. Ainda como trabalhos futuros, propõe-se realizar a análise e a implementação de soluções para assegurar outras propriedades de segurança na SoCIN.

Agradecimentos

Esta pesquisa contou com o apoio do CNPq por meio do Programa Nacional de Microeletrônica (PNM) e do INCT de Sistemas Micro e Nanoeletrônicos (NAMITEC).

Referências

Baron, S., Wangham, M. S. e Zeferino, C. A. (2013). "Security mechanisms to improve the availability of a Network-on-Chip". Em IEEE International Conference On Electronics, Circuits, and Systems (ICECS 2013). Abu Dhabi, p. 609-612.

Baron, S., Wangham, M. S. e Zeferino, C. A. (2014). "Segurança em Redes-em-Chip: conceitos e revisão do estado da arte". Revista de Informática Teórica e Aplicada, Porto Alegre, p. 110-126.

[5] http://www.univali.br/Lists/TrabalhosMestrado/Attachments/748/Sidnei%20Baron.pdf
[6] http://dx.doi.org/10.1109/ICECS.2013.6815488
[7] http://www.capes.gov.br/images/stories/download/avaliacao/Comunicado_004_2012_Ciencia_da_Computacao.pdf
[8] http://seer.ufrgs.br/index.php/rita/article/view/BARON-RITA-VOL21-NR1/29163

Benini, L. e De Micheli, G. (2006). "Networks on chip: technology and tools". New York, Elsevier North-Holland.

Diguet, J., Evain, S., Vaslin, R., Gogniat, G. e Juin, E. (2007). "NoC-centric security of reconfigurable SoC". Em International Symposium on Networks-On-Chip (NOCS), Princeton, IEE Computer Society, p. 223-232.

Fiorin, L., Palermo, G. e Silvano, C. (2008). "A security monitoring service for NoCs. Architecture". Em IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS). New York, ACM, ágp. 197-202.

Fiorin, L., Silvano, C. e Sami, M. (2007). "Security aspects in Networks-on-Chips overview and proposals for secure implementations". Em Euromicro Conference on Digital System Design Architectures, Methods And Tools (DSD). Washington, IEEE Computer Society.

Jerger, N. E. e Peh, L. (2009). "On-Chip Networks: Synthesis Lectures on Computer Architecture". Morgan & Claypool.

Landwehr, C. E. (2001). "Computer security". Em International Journal of Information Security (IJIS). New York, áginasp. 3-13.

Lukovic, S. e Christianos, N. (2010). "Hierarchical multi-agent protection system for NoC based MPSoCs". Em International Workshop on Security and Dependability for Resource Constrained Embedded Systems (S&D4RCES), ACM.

Martin, G. E. e Chang, H. (2003). "Winning the SoC revolution: experiences in real design". New York, Springer, p. 320.

Sepúlveda, M. J. (2011). "Projeto de estruturas de comunicação intrachip baseadas em NoC que implementam serviços de QoW e segurança". Tese, Doutorado em Engenharia Elétrica, Universidade de São Paulo, São Paulo, Brasil, 238 f.

Sepúlveda, M. J., Strum, M. e Chau, W. J. (2009). "Performance impact of QoSS (Quality-of-Security-Service) inclusion for NoC-based systems". Em International Conference on Very Large Scale Integration (VLSI-SOC). Florianópolis, IEEE Computer Society.

Sepúlveda, M. J., Strum, M. e Chau, W. J. (2010). "An hybrid switching approach for NoC-based systems to avoid Denial-of-Service SoC attacks". Em Iberchip Workshop, Foz do Iguaçu.

Stallings, W. (2008). "Criptografia e segurança de redes: princípios e práticas", 4. ed. New Jersey, Prentice Hall.

Texier, M. (2011). "Network-on-Chip security: overview of existing solutions". Disponível em: <http://www.mtexier.com/noc-security>. Acesso em: 12 jul. 2011.

Zeferino, C. A. (2003). "Redes-em-Chip: Arquiteturas e Modelos para Avaliação de Área e Desempenho". Tese, Doutorado em Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil.

Zeferino, C. A. e Susin, A. A. (2003). "SoCIN: a parametric and scalable Network-on-Chip". Em Symposium on Integrated Circuits and Systems (SBCCI), 16. São Paulo, IEEE Computer Society Press, p. 169-174.

Proposta de aprimoramento para o protocolo de assinatura digital Quartz*

Ewerton R. Andrade[1], **Routo Terada**[1]

[1]Departamento de Ciência da Computação (DCC)
Instituto de Matemática e Estatística, Universidade de São Paulo (IME-USP), Brasil

{ewe,rt}@ime.usp.br

Resumo. *O Quartz – criptossistema baseado nos Problemas $\mathcal{MQ}$ ($\mathcal{M}$ultivariate $\mathcal{Q}$uadratic) e IP (Isomorfismo de Polinômios) – é um esquema de assinatura digital pós-quântico resistente a ataques algébricos que visem a recuperação da chave privada. Apesar de sua resistência a ataques algébricos, Joux e Martinet demonstraram que caso o adversário possua um par (mensagem, assinatura) válido, ele conseguirá obter uma segunda assinatura com 2^{50} computações e 2^{50} chamadas ao oráculo de assinatura. Todavia, baseado no Quartz, e visando inviabilizar o ataque de Joux e Martinet, apresentamos um novo esquema de assinatura digital resistente a ataques adaptativos de mensagem escolhida que realizem chamadas ao oráculo aleatório, com um nível de segurança estimado em 2^{112}. Além deste incremento na segurança, nosso criptossistema proporciona um ganho de eficiência no algoritmo de verificação de assinatura e na inicialização dos vetores que serão utilizados pelos algoritmos de assinatura e verificação.*

1. Introdução

Podemos perceber que atualmente, seja conscientemente ou não, uma grande dependência dos sistemas desenvolvidos sob a seara da criptografia foi instaurada em todos nós. Principalmente no tocante dos sistemas criptográficos de chave pública, que são vastamente utilizados na Internet, incluindo-se aí os esquemas de assinatura digital.

No entanto, desde quando Shor em 1997 desenvolveu um algoritmo de tempo polinomial para fatorar inteiros e para calcular o logaritmo discreto num computador quântico (Shor, 1997) a criptografia de chave pública se viu ameaçada e começou a investigar novas fontes de problemas para seus sistemas. Este alarde ocorrera porque, basicamente, os criptossistemas de chave pública usados na atualidade têm sua segurança baseada na intratabilidade dos problemas da fatoração de inteiros, no caso de sistemas RSA, e do logaritmo discreto, em sistemas ElGamal ou de Curvas Elípticas, e tal descoberta tornaria estes sistemas inseguros quando possuíssemos computadores quânticos com a capacidade adequada para implementarmos o algoritmo de Shor.

Uma interessante proposta para enfrentarmos estes desafios é utilização de sistemas MPKC (acrônimo da nomenclatura em inglês que significa Criptossistema de Chave Pública Multivariável), que se apoiam no Problema $\mathcal{MQ}$($\mathcal{M}$ultivariate $\mathcal{Q}$uadratic) para o desenvolvimento ou aprimoramento de sistemas criptográficos de chave pública seguros.

* Texto completo e implementação de referência disponíveis em http://www.ime.usp.br/~ewe/QuartzAprimorado/.

O Quartz é um esquema de assinatura digital baseado no HFEv-, com escolha especial de parâmetros. Sua versão original proposta por Patarin, Courtois e Goubin em 2001 (Patarin et al., 2001) foi atualizada pelos mesmos autores logo em seguida (Courtois et al., 2001), sendo que desde então adotamos esta última como versão original. Este esquema de assinatura foi submetido e aceito no NESSIE[1] (*New European Schemes for Signatures, Integrity and Encryption*). De acordo com os relatórios públicos[2] do NESSIE, o principal trunfo deste esquema são suas assinaturas curtas (apenas 128 bits) e a fundamentação em um problema intratável até mesmo em computadores quânticos (o problema $\mathcal{MQ}$) (Martinet e Supérieure, 2001).

Contudo, o Quartz não foi selecionado para figurar no portfólio final desse projeto de pesquisa. Isto porque – em linhas gerais – o cálculo de suas chaves secretas foi considerado muito lento (Martinet e Supérieure, 2001); por possuir algumas divergências nas especificações de sua implementação (quando confrontado com o requerido pelo NESSIE) (Dottax e Supérieure, 2002); e também por possuir uma arquitetura maleável que permite ao adversário obter uma segunda assinatura, caso ele possua um par (mensagem, assinatura) válido (Joux e Martinet, 2003).

Desta forma, o objetivo de nosso trabalho foi analisar o esquema de assinatura digital Quartz, apresentando, ao final deste estudo, um novo protocolo de assinatura digital baseado nele, porém, com um nível de segurança ainda maior e um algoritmo de verificação mais eficiente. Além disto, foi desenvolvida uma implementação de referência, tanto do modelo original quanto de nosso modelo proposto, para então ser analisada a viabilidade de nosso modelo através da estimativa de segurança e apreciação dos tempos obtidos durante os testes realizados a partir de nossa implementação.

1.1. Contribuições

As principais contribuições deste trabalho são: a apresentação de um novo protocolo de assinatura digital baseado no Quartz, logo, com assinaturas curtas e fundamentado em um problema intratável até mesmo em computadores quânticos; obtenção de um criptossistema resistente a ataques adaptativos que realizem chamadas ao oráculo aleatório, com um nível de segurança estimado em 2^{112}, contra os 2^{50} do protocolo original; constatação de que nosso aprimoramento irá testar até 4.096 vezes menos hipóteses de utilização da chave pública durante a verificação de assinatura, quando comparado com o Quartz Original; implementação do Quartz Original e do Quartz Aprimorado em uma linguagem de programação portável.

2. Quartz Aprimorado

Neste algoritmo será possível notar três grandes mudanças. A primeira delas está no fato de utilizarmos somente uma transformação afim no processo de assinatura das mensagens. Outra mudança está na substituição do SHA-1 pelo SHA-3 no momento de inicializar os vetores de bits que serão utilizados no processo de assinatura e verificação de assinaturas, bem como no momento de gerar as variáveis R e V do algoritmo de assinatura. A terceira grande mudança esta no fato de concatenarmos um *salt* Γ à mensagem M antes de empregarmos a função de hash nesta mensagem.

[1]Similar ao AES, RIPE e CRYPTREC. Maiores detalhes disponíveis em `https://www.cosic.esat.kuleuven.be/nessie/`.

[2]Relatórios disponíveis em `https://www.cosic.esat.kuleuven.be/nessie/reports/`.

2.1. Parâmetros

Em nossa versão aprimorada do Quartz temos definido que: $q = 2$, $d = 129$, $h = 229$, $v = 2$, $r = 5$, $n = 231$ (pois $n \stackrel{def}{=} h + v$), $m = 224$ (pois $m \stackrel{def}{=} h - r$); e a função pública $\mathcal{P}$ – função *trapdoor* – é um mapeamento de 231 bits para 224 bits, ou seja $\mathbb{F}^{231} \mapsto \mathbb{F}^{224}$.

Temos definido, ainda, um parâmetro adicional g, onde g expressa o tamanho do *salt* aleatório Γ que será concatenado à mensagem antes dela servir como entrada para a função hash, ou seja, $g = |\Gamma|$. Lembramos que Sakumoto *et al.*, em seu novo modelo de prova, propuseram a utilização deste *salt* aleatório para uniformizar as assinaturas em esquemas de assinatura digital baseados no HFE (Sakumoto et al., 2011), sendo estimado um tamanho aproximado de $\log(q_{assina}(q_{hash} + q_{assina}))$ bits para que o esquema de assinatura seja considerado seguro (Sakumoto et al., 2011). Sabemos que q_{hash} e q_{assina} correspondem, respectivamente, à quantidade de consultas aos oráculos de hash e assinatura; e que em provas de esquemas de assinatura digital normalmente são considerados $q_{assina} = 2^{30}$ e $q_{hash} = 2^{60}$ (Bellare e Rogaway, 1996). Assim, segue que $g = 96$ bits.

Como estabelecemos que $h = 229$, a extensão do corpo utilizada pelo Quartz Aprimorado fica definida como $\mathbb{F}_{2^{229}} = \mathbb{E}$. Mais precisamente, $\mathbb{E} = \mathbb{F}_2[X]/(X^{229} + X^9 + X^6 + X^5 + X^2 + X + 1)$.

2.2. Assinando Mensagens

Seja M uma mensagem representada por uma cadeia de bits, e S a assinatura obtida desta mensagem. Então, em nosso esquema aprimorado, os procedimentos necessários à obtenção de S devem ser realizados conforme segue:

1. Seja Γ uma cadeia de 96 bits, tal que $\Gamma \in_R \{0,1\}^{96}$;
2. Seja M_0 uma cadeia de 512 bits definida por $M_0 = \text{SHA-3}(M \| \Gamma)$.
3. Sejam H_1 e H_2 duas cadeias de 224 bits: $H_1 = [M_0]_{0 \to 223}$, $H_2 = [M_0]_{224 \to 447}$.
4. Seja $\tilde{S}$ uma cadeia de 224 bits, tal que $\tilde{S}$ seja inicializada com $00...0$.
5. Para $i = 1$ até 2, faça:
   (a) Calcule a cadeia de 224 bits Y definida por $Y = H_i \oplus \tilde{S}$.
   (b) Calcule a cadeia de 512 bits W definida por $W = \text{SHA-3}(Y \| \Delta)$.
   (c) Obtenha a cadeia de 5 bits R definida por $R = [W]_{0 \to 4}$.
   (d) Obtenha a cadeia de 2 bits V definida por $V = [W]_{5 \to 6}$.
   (e) Considerando $F_V(Z) = (Y \| R)$ em Z sobre $\mathbb{E}$:
      i. Se a equação $F_V(Z) = (Y \| R)$ não tiver solução, troque W por SHA-3(W) e retornar ao passo 5c.
      ii. Neste passo a equação $F_V(Z) = (Y \| R)$ tem uma ou mais soluções em $\mathbb{E}$. Logo, temos que $A(1), A(2), ..., A(\delta)$ são as soluções de $F_V(Z) = (Y \| R)$.
      iii. Se $F_V(Z) = (Y \| R)$ tiver apenas uma solução, defina $A = A(1)$. Caso contrário, aplique a função hash em cada uma das soluções, ou seja $I(j) = \text{SHA-3}(A(j))$. Em seguida escolha o $A(j)$ que resulta no menor $I(j)$, considerando a ordenação *big-endian*.
   (f) Calcule a cadeia de 231 bits X definida por $X = s^{-1}\left(\varphi^{-1}(A) \| V\right)$.
   (g) Defina um novo valor para $\tilde{S}$ como sendo $\tilde{S} = [X]_{0 \to 223}$.
   (h) Obtenha a cadeia de 7 bits X_i definida por $X_i = [X]_{224 \to 230}$.
6. A assinatura S é a cadeia de 334 bits definida por $S = \tilde{S} \| X_2 \| X_1 \| \Gamma$.

2.3. Verificando Assinatura

Dadas uma mensagem M – representada por uma cadeia de bits – e uma assinatura S, que neste caso é uma cadeia de 334 bits. Então, os procedimentos que seguem devem ser realizados para verificar se S é ou não uma assinatura válida para M.

1. Seja $\tilde{S}$ uma cadeia de 224 bits definida por $\tilde{S} = [S]_{0 \to 223}$.
2. Sejam X_2 e X_1 duas cadeias de 7 bits: $X_2 = [S]_{224 \to 230}$, $X_1 = [S]_{231 \to 237}$.
3. Seja Γ uma cadeia de 96 bits definida por $\Gamma = [S]_{238 \to 334}$.
4. Seja M_0 uma cadeia de 512 bits definida por $M_0 = \text{SHA-3}(M \| \Gamma)$.
5. Sejam H_1 e H_2 duas cadeias de 224 bits: $H_1 = [M_0]_{0 \to 223}$, $H_2 = [M_0]_{224 \to 447}$.
6. Seja U uma cadeia de 224 bits, tal que U seja inicializada com $\tilde{S}$.
7. Para $i = 2$ até 1, faça:
   (a) Calcule a cadeia de 224 bits Y definida por $Y = G(U \| X_i)$.
   (b) Defina um novo valor para a cadeia de 224 bits U como sendo $U = Y \oplus H_i$.
8. Se U é igual à cadeia $00...0$, aceite a assinatura. Caso contrário, rejeite-a.

3. Análise da proposta

Escolha de Parâmetros

Sabemos que no Quartz, caso o adversário possua um par (mensagem, assinatura) válido, é possível que este adversário obtenha uma segunda assinatura válida com $2^{m/2}$ computações e $2^{m/2}$ chamadas ao oráculo aleatório (Joux e Martinet, 2003). Como nosso aprimoramento não modifica a estrutura geral do Quartz, apenas a adapta; podemos deduzir que tal ataque também é válido para ele. Desta forma, temos que $m \geq 224$ para obtermos um nível de segurança de no mínimo 2^{112} (padrão mínimo exigido para sistemas criptográficos atuais (Barker e Roginsky, 2011)).

Todavia, ao definirmos nossos parâmetros não nos preocupamos somente com o tamanho de m. Isto porque ao estabelecermos parâmetros para instanciar uma função $\mathcal{MQ}$, devemos ter $n > m$ e $n \approx m$ para que o problema permaneça intratável. Também existe o fato de Ding e Schmidt em 2005 terem demonstrado ser possível quebrar o HFEv quando $v = 1$ (Ding e Schmidt, 2005), assim, tomamos o cuidado de escolher um $v > 1$, porém não excessivamente maior. Além disto, escolhemos cuidadosamente o valor de h para que o mesmo fosse primo (fato que também ocorre no modelo original (Courtois et al., 2001; Patarin et al., 2001)); isto porque até hoje não foi apresentada nenhuma criptoanálise que atinja MPKCs que utilizem extensões de corpos com característica igual a um número primo (Feldmann, 2005; Lin et al., 2011; Wolf, 2005).

Desta forma, lembramos que os parâmetros de nosso modelo aprimorado são: $m = 224$, $n = 231$, $h = 229$, $v = 2$, $r = 5$, $q = 2$, $d = 129$ e $g = 96$.

Com estes parâmetros, constatamos dois inconvenientes em nosso aprimoramento. O primeiro deles está no aumento das chaves de nosso criptossistema, pois a chave privada aumenta de 3 Kbytes no Quartz Original para 8 Kbytes em nosso protocolo, sendo que a chave pública salta de 71 Kbytes para 739 Kbytes. O outro inconveniente está na perda de eficiência dos algoritmos de Geração e Chaves e Assinatura (perda que pode ser constatada na Tabela 1 da Seção 5).

Apesar desta perda de eficiência ocorrer, acreditamos que ela não seja tão grave, uma vez que a geração é efetuada apenas uma vez para cada usuário, dentro de um prazo

longo de validade das chaves. Além disto, em determinados cenários, a ineficiência do processo de assinatura não é grave se supormos que o ato de assinar é realizado apenas uma vez para cada mensagem (ou cada Certificado Digital), enquanto a verificação é efetuada por muitos receptores da mensagem (ou do Certificado Digital).

Contudo, aliado ao expressivo ganho no nível de segurança (para maiores detalhes consulte a seção 4), a escolha de parâmetros do nosso esquema aprimorado proporcionou, também, uma melhoria no algoritmo de Verificação de Assinatura. Isto porque testaremos até 4.096 vezes menos hipóteses de utilização da chave pública no momento da resolução da função G, enquanto estamos verificando a validade de uma assinatura.

Substituição do SHA-1 pelo SHA-3

Sabemos que algoritmos de assinatura e verificação são mais rápidos quando aplicados sobre o resultado de uma função hash (y) do que quando aplicado diretamente sobre sua entrada (x) (Terada, 2008). Logo, caso a função hash $H()$ não seja resistente a colisões, um adversário poderia obter uma mesma assinatura para duas mensagens distintas. Para ilustrar o quão prejudicial pode ser utilizar uma função hash inadequada, pensemos em nosso aprimoramento. Suponha que em vez do SHA-3 de 512 bits (que tem sua segurança estimada em 2^{256} (Chang et al., 2012)) utilizássemos o SHA-1. Desta forma, um adversário de nosso criptossistema poderia forjar uma segunda assinatura com aproximadamente 2^{33} operações (Wang et al., 2005) sem atacar diretamente o nosso protocolo.

Portanto, acreditamos que substituir a função SHA-1 pelo SHA-3 seja imprescindível para mantermos nosso esquema aprimorado dentro da segurança estimada. Sendo que, conforme pode ser vislumbrado na Tabela 1, a adesão ao SHA-3 além de ajudar na segurança do Quartz Aprimorado também proporciona um ganho de eficiência no momento de inicializarmos os vetores.

Modificação proposta por Sakumoto *et al.* em 2011

Sakumoto *et al.* desenvolveram um novo modelo de prova, demonstrando que criptossistemas baseados nas *trapdoors* HFE e UOV podem ser existencialmente seguros contra ataques adaptativos de mensagem escolhida (Sakumoto et al., 2011). Entretanto, para utilizarmos este modelo na demonstração de segurança de um criptossistema baseado no HFE, como é o caso do Quartz Aprimorado, necessitamos concatenar um *salt* à mensagem antes de empregarmos a função de hash nesta mesma mensagem afim de obtermos assinaturas uniformemente distribuídas (Sakumoto et al., 2011). Esta modificação acarreta um aumento no tamanho final da assinatura. Em nosso modelo aprimorado este aumento é de 96 bits. Porém, mesmo com este aumento no tamanho final da assinatura, consideramos viável a adesão desta modificação já que nosso esquema aprimorado poderá ser provado como sendo "fortemente infalsificável" em vez de somente "infalsificável", como ocorre no Quartz Original.

Como acreditamos ser dedutível que uma segunda rodada de operações lineares não adicione segurança alguma a um esquema baseado na intratabilidade de equações multivariáveis quadráticas, não nos preocuparemos em dar maiores detalhes sobre a não utilização de duas transformações afim em nosso aprimoramento. Sendo que o leitor pode consultar maiores detalhes sobre o tema em Kipnis et al. (1999), Ding et al. (2006), Bernstein et al. (2009) e Bouillaguet et al. (2011).

4. Estimativa de Segurança

Melhor ataque ao Quartz (HFEv-)

Até o momento não foi desenvolvido nenhum ataque ao Quartz capaz de recuperar a chave privada (inverter a função G) com um esforço menor do que o Ataque por Força Bruta, ou seja, a *trapdoor* HFEv- permanece segura (Bernstein et al., 2009; Ding et al., 2006; Lin et al., 2011).

Apesar disto, em 2010, Bouillaguet *et al.* apresentaram um algoritmo para resolução de equações polinomiais em $\mathbb{F}_2$ que pode ser empregado para solucionar qualquer instância de problema $\mathcal{MQ}$ onde $q = 2$. Com este novo método os autores deste trabalho demonstraram ser possível encontrar todos os zeros de um polinômio de grau d com n variáveis efetuando $d.2^n$ operações binárias (Bouillaguet et al., 2010, Teorema 1). Como este é o algoritmo para resolução de equações polinomiais através da Busca Exaustiva mais rápido da atualidade, então, podemos conjecturar que um adversário conseguirá inverter a função pública G com uma probabilidade estimada em $\epsilon' \leq 1/(d.2^n)$. Desta forma, com os parâmetros adotados em nossa proposta de aprimoramento, temos que $\epsilon' \leq 1/(129.2^{231}) \approx 2^{-238}$.

Estimativa de segurança segundo Sakumoto *et al.*

No apêndice A de Sakumoto et al. (2011), onde consta a demonstração dos Teoremas formulados por Sakumoto *et al.*, os autores mostram que a probabilidade do algoritmo de inversão, simulado pelo oráculo aleatório, encontrar a inversa da assinatura S utilizando chamadas a este oráculo aleatório e a chave pública G é de aproximadamente $\epsilon(1 - (q_{hash} + q_{assina})q_{assina}2^{-g})/(q_{hash} + q_{assina} + 1)$. Desta forma, obtemos $\epsilon = \epsilon'(q_{hash} + q_{assina} + 1)/(1 - (q_{hash} + q_{assina})q_{assina}2^{-t}) \approx \epsilon'.2^{60}$ (Sakumoto et al., 2011, Teorema 2)

Assim, utilizando este Teorema que compõe o novo modelo de prova de segurança proposto por Sakumoto *et al.* e o ϵ' calculado anteriormente, temos que a probabilidade de um adversário recuperar a chave privada do Quartz Aprimorado utilizando um oráculo aleatório é de $\epsilon \approx 2^{-238}.2^{60} \implies \epsilon \approx 2^{-178}$.

Estimativa de esforço para o ataque de Joux e Martinet

Joux e Martinet em 2003 desenvolveram um poderoso ataque ao Quartz; neste trabalho, os autores – baseados em axiomas do Ataque pelo Paradoxo de Aniversário – demonstraram que caso o adversário possua um par (mensagem, assinatura) válido, ele conseguirá obter uma segunda assinatura com $2^{m/2}$ computações e $2^{m/2}$ chamadas ao oráculo de assinatura, com um método que consiste em encontrar a segunda pré-imagem sem se preocupar com a inversão da função pública G (Joux e Martinet, 2003, Seção 3.2).

Desta forma, com os parâmetros adotados em nossa proposta de aprimoramento, temos que através deste ataque o adversário terá que realizar $2^{224/2} = 2^{112}$ computações e $2^{224/2} = 2^{112}$ chamadas ao oráculo de assinatura.

Portanto, para recuperar a chave privada do Quartz Aprimorado será necessário um esforço maior do que 2^{178}, e para derivar uma segunda assinatura através do ataque de Joux e Martinet o adversário terá que efetuar mais de 2^{112} operações e chamadas ao oráculo de assinatura.

			Quartz Original	Quartz Aprimorado
Inicialização dos Vetores	**SHA-1**	**Média (ms)**	158	-
		Intervalo (ms)	121 - 236	-
	SHA-3	**Média (ms)**	-	40
		Intervalo (ms)	-	34 - 57
Geração de Chaves		**Média (s)**	16,9	75,1
		Intervalo (s)	16,5 - 17,7	74,2 - 77,8
Assinatura		**Média (s)**	5,2	19,1
		Intervalo (s)	4,4 - 27,2	18,9 - 20,0
Verificação de Assinatura		**Média (ms)**	3.814	18
		Intervalo (ms)	4 - 3.927	17 - 40
Verificação de Assinatura Falsa		**Média (ms)**	60.074	180
		Intervalo (ms)	52.067 - 62.258	159 - 194

Tabela 1. Tempos obtidos no Brucutu (Intel Xeon E5645 de 2,4 GHz $\times$ 24, com 128 GB de RAM, utilizando o Linux Debian 7.0 (wheezy), OpenJDK 1.6.0_27 IcedTea e Python 2.7.3).

5. Tempos obtidos com a implementação de referência

Acreditamos que a discussão levantada na Seção 3 seja suficiente para justificar os tempos obtidos em nossos testes. Ou seja, a substituição do SHA-1 pelo SHA-3 além de ajudar na segurança do Quartz Aprimorado também proporcionou um ganho de eficiência de aproximadamente $75\ \%$ no momento da Inicialização dos Vetores. Além disso, observamos que os algoritmos de Geração de Chaves e Assinatura têm sua eficiência prejudicada devido a nossa escolha de parâmetros.

Contudo, a Verificação de Assinatura de nosso aprimoramento mostrou-se significativamente melhor do que a do modelo original, sendo possível observar um ganho de $99,5\%$ durante a verificação de assinaturas legítimas e aproximadamente $99,7\%$ quando trata-se de assinaturas falsas.

Frisa-se que os tempos expostos acima foram obtidos a partir de nossa implementação de referência: feita em JAVA, sem paralelismo ou qualquer conjunto de instruções avançadas. Ou seja, todas alterações no desempenho ocorreram em virtude dos parâmetros escolhidos e do novo design proposto em nosso aprimoramento.

6. Considerações Finais

Nossa principal contribuição nesta pesquisa foi a apresentação de um novo protocolo de assinatura digital, baseado em sistemas polinomiais multivariados quadráticos. Como resultado, obtivemos um criptossistema com um nível de segurança estimado em 2^{112}, contra os 2^{50} do protocolo original. Nossa proposta apresenta, ainda, um ganho de eficiência na inicialização dos vetores que serão utilizados pelo algoritmo de assinatura.Além disto, mostramos que no Quartz Aprimorado, durante a resolução da função G, testaremos até 4.096 vezes menos hipóteses de utilização da chave pública, quando comparado com o Quartz Original.

Todavia, observamos que devido os parâmetros escolhidos para nosso criptossistema, houve um aumento significativo no tamanho das chaves, fato que também acarretou uma perda de eficiência nos algoritmos de geração de chaves e assinatura.

Em virtude do tamanho das chaves de nosso aprimoramento, acreditamos que uma possível extensão para nosso trabalho seria pesquisar uma maneira de reluzi-las. Outra possível extensão de nosso trabalho seria buscar uma prova de segurança mais *tight*, no modelo do oráculo aleatório, para protocolos baseados na *trapdoor* HFE.

Agradecimentos

Agradecemos à CAPES pelo apoio financeiro concedido.

Referências

E. Barker e A. Roginsky. NIST Special Publication 800-131A - Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. Relatório técnico, NIST, U.S. Department of Commerce, Washington DC, 2011.

M. Bellare e P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In *EUROCRYPT'96*, páginas 399–416. Springer-Verlag, 1996.

D. J. Bernstein, J. Buchmann e E. Dahmen, editors. *Post-Quantum Cryptography*. Springer, 2009. ISBN 978-3-540-88701-0.

C. Bouillaguet, H. Chen, C. Cheng, T. Chou, R. Niederhagen, A. Shamir e B. Yang. Fast Exhaustive Search for Polynomial Systems in $\mathbb{F}_2$. In *CHES 2010*, páginas 203–218. Springer Berlin Heidelberg, 2010.

C. Bouillaguet, J. Faugère, P. Fouque e L. Perret. Practical Cryptanalysis of the Identification Scheme Based on the Isomorphism of Polynomial with One Secret Problem. In *PKC 2011*, páginas 473–493. Springer Berlin Heidelberg, 2011.

S. Chang, R. Perlner, W. E. Burr, M. S. Turan, J. M. Kelsey, S. Paul e L. E. Bassham. NIST 7896: Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition. Relatório técnico, NIST, U.S. Department of Commerce, Washington DC, 2012.

N. T. Courtois, L. Goubin e J. Patarin. Quartz, an asymmetric signature scheme for short signatures on PC. Primitive spec. and supporting doc. (second revised version). 2001.

J. Ding e D. Schmidt. Cryptanalysis of HFEv and Internal Perturbation of HFE. In *PKC 2005*, páginas 288–301. Springer Berlin Heidelberg, 2005.

J. Ding, J. E. Gower e D. Schmidt. *Multivariate Public Key Cryptosystems*, volume 25 of *Advances in information security*. Springer, 2006.

E. Dottax e É. N. Supérieure. Tweak reviews: ESIGN, RSA-PSS, QUARTZ and SFLASH. NES/DOC/ENS/WP1/018/1. Relatório técnico, 11 2002.

A. T. Feldmann. A Survey of Attacks on Multivariate Cryptosystems. tese de mestrado, University of Waterloo, Ontario, Canada, 2005.

A. Joux e G. Martinet. Some weaknesses in Quartz Signature Scheme. NES/DOC/ENS/WP5/026/1. Relatório técnico, Janeiro 01 2003.

A. Kipnis, J. Patarin e L. Goubin. Unbalanced Oil and Vinegar Signature Schemes. páginas 206–222. Springer Berlin Heidelberg, 1999.

D. Lin, J. Faugère, L. Perret e T. Wang. On Enumeration of Polynomial Equivalence Classes and Their Application to MPKC. Cryptology ePrint Archive, 2011/055, 2011.

G. Martinet e É. N. Supérieure. QUARTZ, FLASH and SFLASH. NES/DOC/ENS/WP3/006/2. Relatório técnico, Março 7 2001.

J. Patarin, N. T. Courtois e L. Goubin. QUARTZ, 128-bit Long Digital Signatures. In *Topics in Cryptology - CT-RSA 2001*, páginas 282–297. Springer B. Heidelberg, 2001.

K. Sakumoto, T. Shirai e H. Hiwatari. On Provable Security of UOV and HFE Signature Schemes against Chosen-Message Attack. In *Post-Quantum Cryptography*, páginas 68–82. Springer Berlin Heidelberg, 2011.

P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

R. Terada. *Segurança de dados: Criptografia em redes de computador*. Blucher, 2ª revisada e ampliada edição, 2008. ISBN 978-85-212-0439-8.

X. Wang, Y. L. Yin e H. Yu. Finding Collisions in the Full SHA-1. In *CRYPTO 2005*, páginas 17–36. Springer, 2005.

C. Wolf. *Multivariate Quadratic Polynomials in Public Key Cryptography*. tese de doutoramento, Katholieke Universiteit Leuven (ESAT), 2005.

Addressing human factors in the design of cryptographic solutions: a two-case study in item validation and authentication*

Fabio Piva[1], **Ricardo Dahab (advisor)**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6176 – 13084-971 – Campinas – SP – Brasil

{fpiva, rdahab}@ic.unicamp.br

Abstract. *Designing secure cryptographic solutions from a purely theoretical perspective is not enough to guarantee their success in a realistic scenario. Many times, the assumptions under which these solutions are designed could not be further from real-world necessities. One particular, often-overlooked aspect that may impact how the solution performs is how the final user interacts with it (i.e., human factors). In this work, we approach this issue by analyzing two well known application scenarios from Information Security research: The e-commerce of digital items and Internet banking.*

1. Introduction

Over the past ten years, selling digital content has become an attractive business, mainly due to the fast increase of consumers' interest in that kind of media. The latest technological trends – such as cheaper and faster broadband internet connections, greater storage space, and devices that provide users with easier access to their data on-the-go – contributed to the development of a solid market for these so-called e-goods. And although several virtual retailers have emerged in order to supply this demand for digital content, most of them have chosen to adopt a real-world-based business model that, however successful for selling physical products, is unsuitable for trading digital goods; it has been observed that physical and digital items are intrinsically too different to be negotiated in the same way [Bottoni et al. 2007].

Another online service commonly engaged by the everyday user is Internet banking (and, more recently, mobile banking). In this scenario, a user is first required to authenticate himself to a remote bank server, to which he will then request some particular task to be performed (such as a money transfer). Other security requirements in this scenario include the guarantee that the user is, in fact, communicating with the bank (i.e., counterpart authentication for the user), and the guarantee that the transaction parameters transmitted to the bank by the user have not been tampered with by an adversary during end-to-end transmission (i.e., transaction authentication).

Even though several solutions have been proposed for authentication scenarios, the Man-in-the-Middle (MitM) attack continues to elude researchers and service providers alike. In fact, and regardless of being extensively researched in the past, the MitM attack remains a real problem in practice [Fang and Zhan 2010, Lee et al. 2010, Hanaek et al. 2008, You et al. 2010, Oppliger 2009] – mostly due to the fact that previous solutions have often been proposed under unrealistic assumptions that do not take into account users' needs or average behavior.

We believe that these two apparently unrelated issues share a common foundation: they are addressed from an essentially flawed design perspective, that fails both to appropriately model the

*PhD. thesis defended on March 28, 2014; Instituto de Computação - UNICAMP. Full text will be soon officially available in http://libdigi.unicamp.br/. In the mean time, it can be downloaded in http://goo.gl/am1F6L.

underlying cause of each problem and to take human factors into account – either as an obstacle to security, or as a tool to provide it. In this thesis, we further extend this argument and approach each individual scenario from a human-centered perspective towards the design of cryptographic solutions.

This document is organized as follows: The remainder of Section 1 includes the original goals of our proposal, as well as accomplished results and contributions. Section 2 refers to our first case of study – the problem of item validation in fair exchange protocols, and how it introduces ambiguities that, ultimately, cause the current business model for buying/selling digital items to allow inaccurate outcomes from a user perspective; the concept of Reversible Degradation – a fair-exchange-based solution for accurate item validation (and therefore, for mitigating mistaken purchases) – is also presented. Section 3 refers to our second case of study – the problem of client-server authentication in *malware*-prone scenarios, which we address in the context of Internet banking applications; in order to provide a suitable solution, we provide a Visual Cryptography method that allows both client and server to authenticate each other (as well as the transaction itself) – even when the client device is hypothetically controlled by a realistic adversary.

1.1. Goals

The main goal of this work is to approach the design of cryptographic solutions while accounting for human factors, as well as evaluating how those factors can be used as assets for guaranteeing security requirements. For that matter, the following specific goals are devised:

1. the study, from a human factors-oriented perspective, of the item validation problem and its impacts in real-world applications in which fairness is required;
2. a survey on special properties of digital items and discussion of their impacts on fair exchange protocols;
3. the proposal of an item validation framework for enabling the fair exchange of particularly hard-to-handle digital items;
4. the implementation, as a proof-of-concept, of an instantiation of the above framework;
5. the study, from a human factors-oriented perspective, of the Internet banking authentication problem and the Man-in-the-Middle attack;
6. the proposal of an authentication solution that addresses the Man-in-the-Middle attack under realistic assumptions;
7. the implementation, as a proof-of-concept, of the above authentication method.

1.2. Main results

- The proposal of the item validation problem as a relevant topic of research in fair exchange e-commerce literature;
- a survey on most commonly observed properties of digital items found in related literature;
- the proposal of the reversible degradation concept as a model for enabling item validation of certain items in fair exchange protocols;
- a proof-of-concept implementation of reversible degradation;
- a discussion of the generic item approach to fair exchange protocol design and proposal of a non-generic, item-aware approach to the process;
- the proposal and proof-of-concept implementation of a transaction authentication method based on Visual Cryptography.

1.2.1. Publications

- Full paper (first author), submitted: "*A secure method for transaction authentication in malware-prone scenarios.*", Submitted to: Financial Cryptography and Data Security, 2015. Isla Verde, Puerto Rico.
- Journal article (first author): "*E-commerce of digital items and the problem of item validation: introducing the concept of reversible degradation.*", In: Journal of Applicable Algebra in Engineering, Communication and Computing, 2013. Springer-Verlag Berlin Heidelberg.
- Full paper (first author): "*Modern fair exchange protocol design: dealing with complex digital items.*", In: XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2013, Manaus/Brazil.
- Extended abstract (first author): "*Using systematic error correcting codes for reversible degradation of multimedia content.*", In: 18th International Conference on Applications of Computer Algebra (ACA), 2012. Sofia/Bulgaria.
- Full paper (first author): "*E-commerce and fair exchange - The problem of item validation.*", In: International Conference on Security and Cryptography (SECRYPT), 2011. Seville/Spain.
- Technical report (first author): "*E-commerce and fair exchange: the problem of item validation.*", Institute of Computing, University of Campinas, 2011. Campinas/Brazil.

1.2.2. Submitted patent requests

- *Degradação Reversível: um método para validação segura de itens digitais em protocolos de comércio eletrônico.* Content number 775, Agência de Inovação Inova, Unicamp;
- *Método de Criptografia Visual para compartilhamento de segredos múltiplos com transparência reutilizável.* Content number 772, Agência de Inovação Inova, Unicamp.

2. E-commerce of digital items with Reversible Degradation

In this section we briefly describe our fist case of study, namely the problem of item validation in the context of e-commerce of digital items.

2.1. Fair exchange protocols and the problem of item validation

Fair exchange protocols were proposed by Asokan [Asokan 1998] as a solution to the problem of two mutually distrusting parties interested in exchanging digital items atomically. Many variations of Asokan's original protocols have since been studied [Ray 2002, Alaraj and Munro 2007], but most of them were too cumbersome or required too many resources to be considered practical for real-world applications. As a result, most current e-commerce stores do not implement fair exchange in their business models; a simple web search reveals several Apple's iTunes Store user complaints about mistaken music files being purchased due to inaccurate description of the products; also, the Digital Downloads section on Amazon.com contains several customer comments on the same subject.

Such problems relate to an essential, but not sufficiently explored, aspect of fair exchange protocols: the **item validation** step [Bottoni et al. 2007]. The original definition of fairness states that "*an exchange is fair if at the end of the exchange, either each player receives the item it expects*

or neither player receives any additional information about the other's item" [Asokan 1998]. For that end, aside from ensuring the atomicity of the exchange, the protocol must specify when and how a party can check whether the item she has just received (or is about to receive) is the one she desires. This is, however, a delicate process that may be influenced by the characteristics of the items being exchanged, by the available resources and by the structure of the protocol itself.

We approach these inherent issues in a top-down fashion (i.e., by initially taking into account the structural and semantic nature of the items that should be exchanged in the protocol), which allows us to provide a set of guidelines that can assist the modern fair exchange protocol designer to build realistic solutions for real-world applications. We call this design approach **item-aware protocol design**, as opposed to the *generic item protocol design* approach introduced in Asokan's original work and followed by many authors after him. As far as generic item protocol design is concerned, the exchanged items are seen as generic bit streams with few or no particular properties of relevance to protocol design. We believe, however, that in most current contexts, items do have inherent complexity that may interfere with transactions, and exhibit characteristics that either make the exchange easier, or become obstacles for enforcing successful (fair) outcomes.

2.2. The Reversible Degradation concept

The reversible degradation concept was proposed as a solution to the problem of item validation. In the currently-established business model for buying and selling digital content, buyers are required to first search an online store catalog for the product they intend to acquire, and then decide – usually by reading a textual description containing information about the product – if that is in fact the item they desire. Unfortunately, this would only be suitable if the available descriptions of an item were able to provide the buyer with every relevant information one might need before purchasing it.

Since buyers may have different, unpredictable expectations/needs concerning an item, and given the complex nature of many currently available digital products, suitable (univocal) descriptions are usually hard to provide – which often leads to unfair outcomes in which the buyer ends up receiving a product that does not satisfy his/her needs [Bottoni et al. 2007]. To worsen the problem, since digital items are easy to copy, such products are usually excluded from most online retailers' return/refund policies [Amazon Legal Department 2005]. In summary, if a purchase transaction results in an undesired/unsatisfying digital product being delivered to the buyer – which is likely to happen in the current description-based e-commerce model – there is little the customer can do to correct the situation. This is not only bad for the buyer, but also for the store in the long term – since it undermines customer satisfaction and, by extension, trust on e-commerce as a whole.

To further illustrate the problem of item validation, consider the following example: A $Buyer$ intends to buy a particular image file – the portrait of model Lena Söderberg – from an online picture retailer $Seller$. $Buyer$ finds on $Seller$'s website a product description $desc(i)$ that apparently satisfies his mental image of i: *Portrait of Lena Söderberg, hat, bust, plumes, PNG image, 256x256 resolution.* After reading $desc(i)$, $Buyer$ decides to purchase it. By doing so, $Buyer$ is provided with access to the content – an image file that satisfies every aspect of description $desc(i)$, but which might not be the desired file. We illustrate in Figure 1 three similar items that can be described by $desc(i)$, and as such, would be candidates for delivery to $Buyer$ after payment.

The main idea behind reversible degradation is to transform the item in such a way that it

Figure 1. Three different items that show pictures of the model Lena Söderberg.

becomes clearly deteriorated (*degraded*), but without depriving it from its main functional (in this case, perceptual) characteristics. This allows the owner of such item to release it to the interested counterpart before payment is made, in order to enhance the chances of a successful validation. If $Buyer$ is satisfied with the validation performed over the degraded version of the item, he then pays for a key that reverts the degradation process and recovers the item to its full, original quality. Figure 2 illustrates the reversible degradation concept.

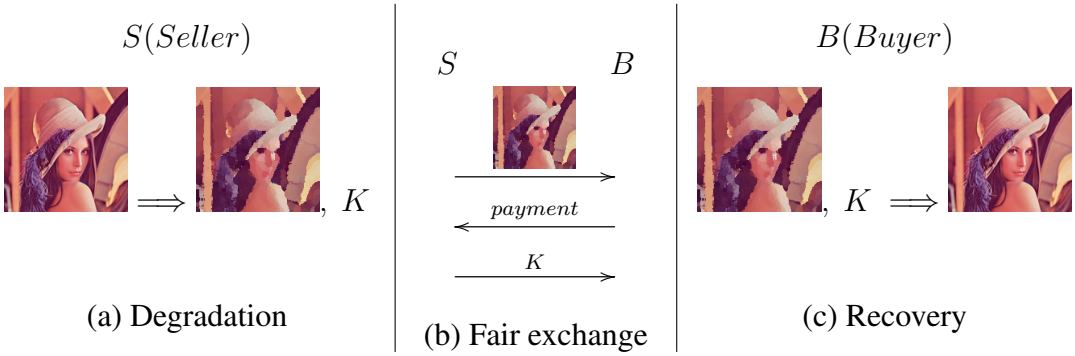


Figure 2. Reversible degradation concept description.

As a proof-of-concept, we designed and implemented a reversible degradation method based on systematic error correcting codes (SECCs) [Reed and Solomon 1960], due to their message preserving property[1]. This property is crucial for the implementation of our reversible degradation technique, since it allows the degraded copy of the item to partially maintain its perceptual functionalities and the degradation level to be performed in a controlled fashion.

Our proposed technique addresses the validation of frame-structured multimedia items consisting of perceptual information (such as MPEG Audio Layer III (MP3) files and PGM/PPM image files, for instance), and is described in two methods: A *degradation method*, which allows an arbitrary amount of degradation to be added to the item and which outputs not only a degraded version of the item, but also a restoring key for reversing the process; and a *recovery method*, which allows the full recovery of the original item from its degraded copy, provided that the restoring key is known.

3. Internet banking authentication with Visual Cryptography

As Internet banking becomes increasingly popular among customers, its potential as target for fraud has also increased [Trend 2013]. In recent years, e-banking fraudsters' activities have matured both quantitative and qualitatively, often relying on previously deployed malicious software

[1]SECCs are able to encode a k-symbol long message m into an n-symbol long codeword, $k < n$, in which the first k symbols correspond to the same symbols that compose m; the remaining $n - k$ symbols are regarded as *parity symbols*. That is, for any message $m = (m_1, m_2, m_3, ..., m_k)$, the resulting codeword $c = (m_1, m_2, m_3, ..., m_k, p_1, p_2, ..., p_{n-k})$.

artifacts (*malware*) that show unprecedented levels of sophistication [Andriesse et al. 2013]. Once the adversary manages to deploy an instance of *malware* on the victim's device, ensuring any level of security becomes a hard, if not impossible, task.

To the extent of our knowledge, and even though extensive research has been conducted on the subject [Fang and Zhan 2010, Lee et al. 2010, Hanaek et al. 2008, Oppliger 2009], no practical approach is known to effectively prevent MitM attacks under the modern adversary model [Oppliger 2009] (i.e., when the user starts the communication from an assumedly-controlled-by-*malware* device). The most common, arguably successful approaches often rely on multi-channel [You et al. 2010] or multi-factor [Aloul et al. 2009] authentication mechanisms. However, even those solutions are recognized as ineffective against an adversary capable of compromising users' devices [Lemos 2009, Schneier 2005, Schneier 2009], since they still allow a motivated adversary to subvert genuine user-initiated sessions into fraudulent ones.

We propose a two-factor/two-channel authentication solution that relies on Visual Cryptography (VC) [Naor and Shamir 1994], which introduces the concept of **non-computational authentication channel** as a solution towards the stronger transaction authentication requirement. This *malware*-free channel enables the user to handle critical (i.e., cryptographic) steps in the protocol in an "out-of-device" fashion, without relying on expensive or *malware*-prone equipment.

Our approach is novel in the following aspects: 1) it does not require the user's device to be uncompromised by *malware*; 2) it satisfies both mutual user and transaction authentication requirements without storing credentials in the user-side (i.e., compromisable) devices; 3) it is cost-effective in comparison to currently implemented solutions; 4) it remains robust even in hostile (yet realistic [Trend 2013]) scenarios where a user's device is hijacked by *malware*; and 5) it effectively protects user transactions against *malware*-enabled attacks (such as MitM-enabled channel breaking and credential stealing, for instance) and certain kinds of social engineering attacks (such as *phishing*). We present our solution in detail and in a step-by-step, constructive approach, also providing insights regarding security, usability and logistical aspects.

3.1. A practical Visual Cryptography method for authentication

Visual Cryptography (VC) was proposed by Naor and Shamir [Naor and Shamir 1994] as a secret sharing method that combines perfect ciphers and perceptual, non-computational information decryption. By subdividing a visual secret into a set of seemingly-randomized bitmap images (i.e., *shares*), a dealer is able to allow a group of parties to recover that secret only when a sufficiently-large subset of those parties agrees to do so. Since the recovery does not require any computation – relying only on the visual capabilities of share holders instead – VC schemes provide a simple and intuitive alternative for secret sharing among two or more parties, which is of particular interest in contexts where sensitive information is to be shared with users with little or no technical knowledge, or with little or no available computational power. In our context of application, this aspect also becomes a promising element for performing secure transactions between a client and server, where the client is supposed to be running on an untrusted device. Figure 3 illustrates the basic two-party VC scenario, in which two shares reveal a numerical secret when stacked together.

Since its proposal, few attempts have been made in adapting VC to authentication scenarios [Naor and Pinkas 1997], only with modest success; this is due the fact that Visual Cryptography incorporates one-time-pad aspects, thus disallowing the inclusion of any previously-produced share in a new set, meant to reveal a different secret.

Our proposed method allows the production of second shares that can be overlaid with the

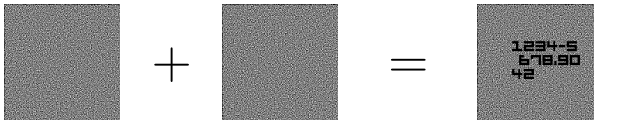
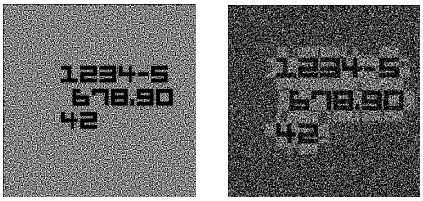


Figure 3. Secret recovery by overlaying a pair of shares in the classical VC scheme.

same first share, so that multiple transactions can be authenticated by an authorized user without the need of any sensitive computation to be performed by the untrusted device. For that purpose, we devise a novel pixel selection algorithm specifically designed to allow the reuse of any pixel (i.e., pixels located at any particular coordinates in the share) for a reasonable number of times – as opposed to previous VC authentication approaches, which aim at subdividing the master share into smaller regions and at avoiding the use of any particular region more than once [Naor and Pinkas 1997]. The main idea behind this selection algorithm is that, by allowing some loss of contrast and resolution on the recovered secret, we become able to significantly enhance the selectable pixels corresponding to each position of the share – thus effectively reducing the amount of information gained by an adversary through statistical cryptanalysis. Figure 4 illustrates a comparison between overlaid shares in the classical ((a), non-suitable for authentication) VC method [Naor and Shamir 1994] and in our proposed ((b), suitable for authentication) method.



(a) (b)

Figure 4. Recovered secret images for comparison.

Since knowledge of the correct first share is required for both visually retrieving the secret and generating valid authentication shares, the client is assured, upon overlaying his first share with a server-generated authentication share and seeing intelligible information, that a legit connection to the server has been established (user authentication for the client). If the revealed information includes the correct parameters regarding the transaction to be executed by the server, the client is also assured that no unauthorized tampering took place before the transaction request reached the server (transaction authentication for the client). As for the server, if the client replies with the correct one-time-password – after the authentication share has been sent – then the server is assured that the client is indeed the originator of that transaction request (user authentication for the server). Finally, the server is also assured that, provided that the client was able to retrieve the correct one-time-password, he must also have been able to verify the transaction parameters that were received by the server and, upon replying with the correct one-time-password, confirms them as accurate (transaction authentication for the server).

References

[Trend 2013] (2013). The Invisible Web Unmasked: TrendLabsSM 3Q 2013 Security Roundup. Technical report.

[Alaraj and Munro 2007] Alaraj, A. and Munro, M. (2007). An e-Commerce fair exchange protocol for exchanging digital products and payments. *2nd International Conference on Digital Information Management (ICDIM).* , 1:248–253.

[Aloul et al. 2009] Aloul, F., Zahidi, S., and El-Hajj, W. (2009). Multi Factor Authentication Using Mobile Phones. *International Journal of Mathematics and Computer Science*, 4(2):65–80.

[Amazon Legal Department 2005] (2005) Amazon Legal Department. Product Return Policies: Digital Content. Online; accessed 25-February-2014.

[Andriesse et al. 2013] Andriesse, D., Rossow, C., Stone-Gross, B., Plohmann, D., and Bos, H. (2013). Highly resilient peer-to-peer botnets are here: An analysis of Gameover Zeus. In *8th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 116–123.

[Asokan 1998] Asokan, A. (1998). *Fairness in Electronic Commerce*. PhD thesis, University of Waterloo.

[Bottoni et al. 2007] Bottoni, A., Dini, G., and Stabell-Kulø, T. (2007). A methodology for verification of digital items in fair exchange protocols with active trustee. *Electron Commerce Res*, 7(2):143–164.

[Fang and Zhan 2010] Fang, X. and Zhan, J. (2010). Online Banking Authentication Using Mobile Phones. *2010 5th International Conference on Future Information Technology*, pages 1–5.

[Hanaek et al. 2008] Hanaek, P., Malinka, K., and Schafer, J. (2008). E-banking security - comparative study. In *42nd Annual IEEE International Carnahan Conference on Security Technology*, pages 326–330. IEEE Computer Society.

[Lee et al. 2010] Lee, Y. S., Kim, N. H., Lim, H., Jo, H., and Lee, H. J. (2010). Online banking authentication system using mobile-OTP with QR-code. *5th International Conference on Computer Sciences and Convergence Information Technology*, pages 644–648.

[Lemos 2009] Lemos, R. (2009). Real-Time Hackers Foil Two-Factor Security. `goo.gl/cPecRj`. Online; accessed 03-July-2014.

[Naor and Pinkas 1997] Naor, M. and Pinkas, B. (1997). Visual authentication and identification. In Kaliski, B., editor, *Advances in Cryptology (CRYPTO '97)*, volume 1294 of *Lecture Notes in Computer Science*, pages 322–336. Springer Berlin / Heidelberg.

[Naor and Shamir 1994] Naor, M. and Shamir, A. (1994). Visual Cryptography. In *Eurocrypt 94*, pages 1–12.

[Oppliger 2009] Oppliger, R. (2009). Internet Banking: Client Side Attacks and Protection Mechanisms. (June):27–33.

[Ray 2002] Ray, I. (2002). Fair exchange in e-commerce. *ACM SIGecom Exchanges*, 3(2):9–17.

[Reed and Solomon 1960] Reed, I. S. and Solomon, G. (1960). Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304.

[Schneier 2005] Schneier, B. (2005). Two-factor authentication: Too little, too late. *Commun. ACM*, 48(4):136.

[Schneier 2009] Schneier, B. (2009). Hacking Two-Factor Authentication. `goo.gl/BcNQPY`. Online; accessed 03-July-2014.

[You et al. 2010] Han-Na You, Jae-Sik Lee, Jung-Jae Kim, and Moon-Seog Jun. (2010). A study on the two-channel authentication method which provides two-way authentication in the Internet banking environment. In *5th International Conference on Computer Sciences and Convergence Information Technology*, pages 539–543. IEEE.

Avaliação Resiliente de Autorização $UCON_{ABC}$ para Computação em Nuvem

Arlindo L. Marcon Jr., Altair O. Santin (*orientador*)
Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR) – Curitiba, PR – Brasil
{almjr, santin}@ppgia.pucpr.br

Resumo. *O consumidor de nuvem necessita controlar o consumo individual de seus usuários. O modelo $UCON_{ABC}$ permite avaliações periódicas, executando a reavaliação contínua dos atributos de autorização. Porém, o $UCON_{ABC}$ não foi projetado para o contexto da nuvem. Este trabalho mostra que é possível prover resiliência ao processo de reavaliação de autorização do $UCON_{ABC}$. O protótipo como prova de conceito mostra que é possível prover elasticidade às entidades responsáveis por avaliar e contabilizar os atributos de uso.*

Abstract. *The cloud consumer needs to control the individual consumption of their users. The $UCON_{ABC}$ model allows periodic evaluations, performing continuous reevaluations of authorization attributes. However, the $UCON_{ABC}$ was not designed for the context of the cloud computing. This work shows that it is possible to provide resilience to the $UCON_{ABC}$ authorization revaluation. The proof-of-concept prototype shows that providing elasticity is feasible for evaluating entities and accounting attributes.*

1. Introdução

A computação em nuvem fornece serviços que podem ser contratados conforme a demanda do consumidor [Hayes, 2008]. As principais entidades envolvidas nesse contexto são: *i) consumidor*, contrata os serviços para atender seus usuários; *ii) usuário*, sujeito que utiliza os serviços - *i.e.* o usuário final; *iii) provedores*, fornecem os serviços para a nuvem computacional - *e.g.*, infraestrutura, plataforma [Marcon Jr. *et al.*, 2010].

Os provedores de serviço devem honrar os contratos (*Service Level Agreements - SLAs*) firmados com o consumidor, controlando o acesso em nível de usuário e administrativo [Emeakaroha *et al.*, 2011]. Adicionalmente, os provedores deveriam monitorar os recursos alocados e oferecer um esquema de gerenciamento eficiente para seus consumidores [CSA, 2011]. Cada usuário do consumidor pode originar diferentes cenários de carga para a nuvem (*e.g.*, processamento, armazenamento), utilizando recursos de vários provedores. Neste caso, a abordagem tradicional (*i.e.*, configuração estática de políticas nos serviços) pode caracterizar a subutilização de recursos em um provedor e a sobrecarga destes em outro (*i.e.*, não há como prever as demandas). Em um contexto ideal, as políticas de uso deveriam ser frequentemente avaliadas visando detectar oscilações no consumo. A monitoração periódica dos recursos asseguraria a utilização uniforme dos serviços, sem prejudicar o acesso do usuário.

Neste trabalho, agentes de monitoramento coletam informações de consumo enquanto o controle de uso (*i.e.*, $UCON_{ABC}$) avalia continuamente as autorizações [Park *et al.*, 2004]. A continuidade do uso é concedida conforme a reavaliação das políticas. O uso pode ser entendido como operações de escrita em um objeto (*e.g.*, um arquivo) ou o consumo de recursos (*e.g.*, ciclos de *CPU*). A frequência da coleta de atributos reflete diretamente no período de tempo em que o usuário pode estar violando uma política, situação esta que caracteriza uma exceção (*i.e.*, disparidade entre a autorização concedida e a política vigente). Evidentemente, é desejável aplicar o menor intervalo de

Este trabalho é parcialmente financiado pelo Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC/MCTI), contrato 1313 e CNPq, processos: 478285/2011-6 e 310671/2012-4.

tempo possível para a obtenção dos atributos de uso e a reavaliação das políticas. Esta proposta mostra um cenário em que os períodos de inconsistência de autorização (*i.e.*, situação na qual o usuário está em condição de exceção) são mitigados.

O trabalho está organizado em: seção 2, aborda a nuvem computacional; seção 3, trata do controle de uso; seção 4, apresenta alguns trabalhos relacionados; seção 5, contém a proposta; seção 6, protótipo e testes; por fim, a seção 7 elenca as conclusões.

2. Computação em Nuvem

A nuvem pode ser descrita pelos seguintes itens [Mell *et al.*, 2009]: *i)* auto-atendimento sob-demanda; *ii)* serviços amplamente disponíveis na rede; *iii) pool* de recursos computacionais; *iv)* rápida elasticidade; *v)* contabilização de consumo. Esse modelo é formado por um conjunto de serviços classificados em: *i) Software como um Serviço - SaaS*: representa o *software* que o consumidor utiliza; *ii) Plataforma como um Serviço - PaaS*: permite ao consumidor instalar aplicações e gerenciar as configurações subjacentes; *iii) Infraestrutura como um Serviço - IaaS*: provê recursos computacionais (*e.g.*, processador, memória) para a execução do sistema operacional (*e.g.*, *Linux*).

3. Controle de Uso

A abordagem mais adequada para ambientes dinâmicos é o controle de uso $UCON_{ABC}$ [Park *et al.*, 2004]. Este modelo reavalia as autorizações periodicamente, levando em consideração a mutabilidade dos atributos. As autorizações seguem o modelo tradicional de avaliação e concessão de direitos (*e.g.*, leitura, escrita). Porém, o $UCON_{ABC}$ faz a avaliação continua, tendo em vista que os atributos do usuário ou objeto (*e.g.*, serviço) podem ser alterados à medida que o acesso é executado. Assim, os atributos de consumo podem ser atualizados antes ou durante a utilização do objeto, necessitando de uma avaliação de autorização antes (*pre*) e durante (*ongoing*) o uso. A avaliação *pre* é clássica, enquanto a avaliação *ongoing* é necessária devido à mutabilidade dos atributos.

4. Trabalhos Relacionados

A proposta de Zhang [Zhang *et al.*, 2009] auxilia os provedores a aumentarem a flexibilidade dos serviços oferecidos na nuvem (*i.e. SaaS*). O trabalho adota o esquema de segurança tradicional, desconsiderando a flexibilidade dos demais níveis de serviço (*e.g. PaaS*, *IaaS*). Para Lim [Lim *et al.*, 2009], políticas munidas de atributos fornecidos pelo provedor auxiliam o consumidor a administrar a utilização dos recursos alocados. O artigo expõe a necessidade do consumidor em gerenciar a cota contratada. O trabalho de Bertram [Bertram *et al.*, 2010] faz o mapeamento de atributos do ambiente em políticas de controle. Porém, a proposta não deixa claro como o processo é executado.

A proposta de Tavizi [Tavizi *et al.*, 2012] aplica o tratamento de obrigações $UCON_{ABC}$ em ambientes de nuvem. O trabalho de Danwei [Danwei *et al.*, 2009] apresenta um módulo de negociação para aumentar a flexibilidade do sistema de controle de acesso. Ambas as pesquisas desconsideram a periodicidade do sistema de monitoramento de atributos, item que reflete diretamente na reavaliação de autorização.

5. Avaliação Resiliente de Autorização $UCON_{ABC}$

A proposta estende o modelo de autorização $UCON_{ABC}$, provendo resiliência a reavaliação das políticas de uso [Marcon Jr. *et al.*, 2013]. Resiliência, nesta abordagem, significa prover ao modelo a habilidade de tratar algumas situações de exceção que ocorrem com os atributos de autorização do usuário. Porém, mantendo as cotas de consumo do serviço dentro dos parâmetros definidos no *SLA* (Figura 1; *evento* SLA_{CO}). O domínio consumidor escreve as políticas de uso para seus usuários (*evento* U_{AQ}),

definindo os atributos de autorização individuais (*i.e.*, as cotas de uso de serviço). A cota é utilizada durante o processo de reavaliação de políticas em substituição aos atributos de autorização. O objetivo deste esquema é flexibilizar as políticas de uso, quando possível, lidando com situações de exceção de autorização.

Atributos que contabilizam o consumo do usuário são obtidos do provedor através de agentes de monitoramento (*Agm*; *evento* A_{UU}; Figura 1). A resiliência para o processo de reavaliação de autorização contínua está definida somente se o SLA_{CO} menos a soma que contabiliza todos os atributos de uso dos usuários é maior que *t*. A constante *t* é uma cota reserva definida pelo consumidor para um serviço. Isto significa que, quando a soma de consumo total dos usuários (*sctu*) estiver próxima do limiar definido por "SLA_{CO} - *t*", um novo *SLA* deverá ser negociado para evitar uma violação de contrato.

Um cenário envolvendo um sistema de arquivos é utilizado para explicar como a cota é utilizada para prover resiliência ao processo de autorização $UCON_{ABC}$. Neste contexto (Figura 1) a resiliência está representada por linhas pontilhadas para os atributos de uso e de autorização. Considerando um consumidor que negocia um contrato (*evento* SLA_{CO}) para um serviço de armazenamento com *600GB*, sendo *t* = *100GB*. O consumidor escreve políticas que definem *userA: 200GB*, *userB: 200GB* e *userC: 100GB*. Quando o usuário solicita acesso ao serviço (*eventos* AC_A e AC_B), o guardião (*GS*) envia uma solicitação de avaliação de autorização para o monitor de referência $UCON_{ABC}$ (*MRU*; *evento PER*). O *MRU* configura a cota de consumo para o usuário (*i.e.*, o valor inicial da cota é igual ao atributo de autorização definido pelo consumidor: *userA: 200GB*, *userB: 200GB* e *userC: 100GB*), fornecendo a permissão de consumo para o mecanismo que executa o controle de acesso (*evento PDE*).

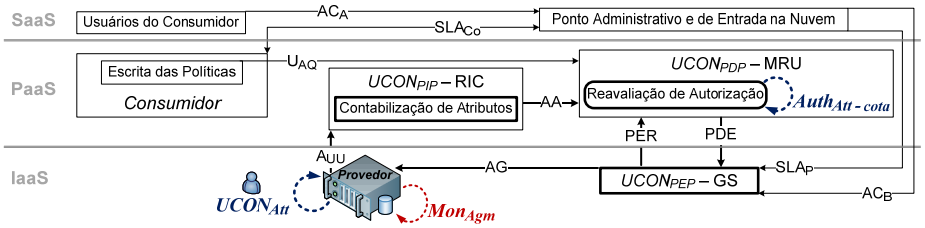


Figura 1. Modelo de autorização resiliente $UCON_{ABC}$

Após ter o acesso liberado pelo guardião (*evento AG*), o usuário começa a utilizar o serviço. Em seguida, o agente de monitoramento (*Agm*) envia os atributos de uso de cada usuário (*e.g.*, *userA: 190GB, userB: 10GB* e *userC: 0GB*) para o repositório de informações de contexto (*RIC*). Passado algum tempo, o guardião (*GS*) solicita uma reavaliação de autorização. Durante a reavaliação (autorização *ongoing*), o monitor de referência (*MRU*) compara os atributos de cada usuário (*evento AA*) com a cota (*i.e.*, mecanismo de resiliência). Neste caso, o armazenamento utilizado por cada usuário está abaixo das cotas predefinidas. Novamente, após um período de tempo, o agente envia outra vez os atributos de uso (*e.g.*, *userA: 250GB, userB: 20GB* e *userC: 10GB*) para o repositório (*RIC*) e uma reavaliação de autorização é necessária. Durante a reavaliação, o *MRU* percebe que os atributos do *userA* estão excedendo a cota predefinida (*i.e. userA: 200GB*). Neste momento, a condição de soma de consumo para os usuários (*sctu*) é de *280GB*. Como o "SLA_{CO} - *t*" admite *500GB* de armazenamento, a cota para o usuário em situação irregular é automaticamente expandida para *userA: 250GB*.

Periodicamente, o agente (*Agm*) envia os atributos de cada usuário (*e.g.*, *userA: 240GB, userB: 160GB* e *userC: 100GB*) para o repositório (*RIC*). Enquanto isto, o guardião (*GS*) continua solicitando as reavaliações de autorização para o monitor de referência (*MRU*). Neste processo, o *MRU* percebe que os atributos do *userA* estão

abaixo da cota expandida, porém, a condição de soma de consumo (*sctu*) é de *500GB* de armazenamento. Neste momento, o processo de reavaliação detecta que o espaço de armazenamento, subtraído da cota reserva (SLA_{CO} - *t*) foi totalmente utilizado. Adicionalmente, a cota do *userA* está excedendo a política, sendo que a parcela de armazenamento extra deveria ser equiparada com o atributo de autorização original - *i.e. userA: 200Gb*. Se, na próxima reavaliação, os atributos do *userA* se mantiverem além da cota, e a soma dos atributos (*sctu*) estiver próxima do SAL_{CO}, o *userA* vai estar em uma condição de exceção (*i.e.*, o usuário utilizou mais de *200GB*, não sendo possível manter a resiliência). Caso contrário, a cota do *userA* poderia ser alterada novamente.

A exceção ocorre porque entre os períodos de reavaliação, o usuário continua consumindo o serviço. Essa situação também pode ser desencadeada pela resiliência do modelo, o qual redefine a cota para o valor original do atributo de autorização. O modelo proposto provê um equilíbrio automático entre o limite estabelecido na política e a quantidade definida no *SLA*. O esquema de reavaliação é resiliente para o processo de autorização, explorando a ociosidade na utilização dos serviços - desde que o *sctu* esteja abaixo do limiar "$SLA_{CO} - t$". Nas abordagens tradicionais, a autorização é avaliada somente no inicio da utilização (*pre*), porém isto pode gerar inconsistências entre a autorização concedida e a política de uso vigente. Mesmo a reavaliação de autorização durante o acesso (*ongoing*) não garante que um consumo autorizado não vai violar a política. Isto ocorre porque as condições de exceção acontecem entre os períodos de reavaliação. Nesta proposta, estes períodos são menores do que nas abordagens tradicionais, dependendo apenas do intervalo entre as reavaliações de autorização.

5.1. Arquitetura do Modelo

A proposta utiliza um ambiente intermediário (Ambiente Federado - *AF*; Figura 2) para facilitar a interação entre as entidades e para prover um *PaaS* seguro para a execução do modelo de autorização resiliente. Os Provedores de Serviço (*PS*) e consumidores filiam-se ao ambiente de nuvem através do *Broker*. O *Broker* faz a intermediação da oferta de serviços (*IaaS*), negocia os *SLAs* com os consumidores, e executa o redirecionamento dos usuários do consumidor (*DC*) para o endereço do provedor definido no *SLA*.

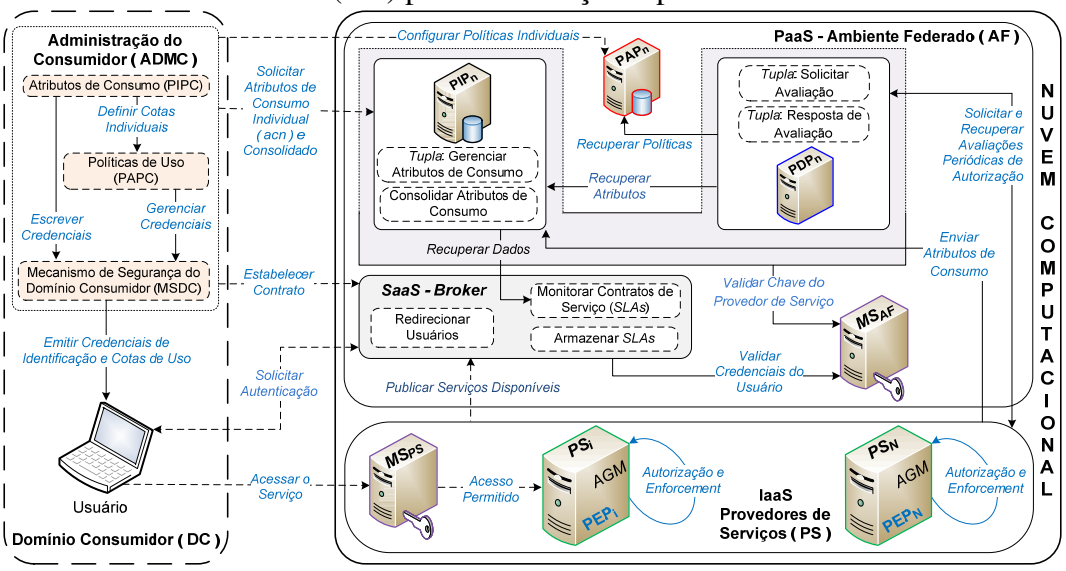


Figura 2. Visão geral da arquitetura proposta

Para cada serviço o provedor envia ao *Broker*: *a) SLA*: define os montantes que podem ser repassados aos consumidores; *b) descrição do serviço*: documento que o

usuário do consumidor utiliza para acessar o serviço. O consumidor estabelece o *SLA* com o *Broker* e define as políticas para seus usuários. As regras de uso são configuradas no ponto de administração de políticas (*PAP*; Figura 2) da federação. Cada *SLA* fornece ao consumidor um conjunto de serviços para gerenciar o ambiente (*e.g.*, armazenamento de atributos (*PIP*) e políticas (*PAP*)). A administração do consumidor (*ADMC*) é responsável por escrever as credenciais de autenticação para seus usuários no Mecanismo de Segurança do Domínio Consumidor (*MSDC*). Estas credenciais permitem aos usuários interagirem com o *AF* e consumirem os serviços instanciados no provedor.

Antes do usuário utilizar sua cota de uso, esse deve enviar ao *Broker* a credencial de autenticação assinada pelo consumidor. O *Broker* é quem redireciona o usuário para o serviço desejado. A solicitação de acesso recebida pelo provedor é interceptada pelo guardião (*PEP*; Figura 2) e avaliada pelo monitor de referência (*PDP*) instanciado no Ambiente Federado (*AF*). Esta avaliação utiliza as informações de contabilização de consumo do usuário armazenadas no repositório de informações de contexto (*PIP*). O repositório é atualizado por agentes de monitoramento (*AGM*) instanciados nos provedores de serviço. Após o consumidor configurar as políticas de uso, e o usuário iniciar a utilização do serviço, o *PIP* passa a armazenar atributos de consumo referentes a cada usuário acessando o serviço. Periodicamente os dados são consolidados no *PIP* para avaliar se o SLA_{CO} está sendo respeitado. No Domínio Consumidor, estes dados são analisados para decidir se as políticas de uso individuais precisam ser readequadas.

Considerando avaliações anteriores, o modelo *outsourcing* se mostrou mais adequado a nuvem devido à dinamicidade deste ambiente [Marcon Jr *et al.*, 2009]. Com esse modelo evitam-se as inconsistências causadas pelo *cache* de políticas no provedor. A proposta emprega serviços de espaço de *tuplas*, sendo desacoplada no tempo e espaço. Estes serviços são utilizados para armazenar as solicitações de reavaliação de autorização (enviada pelos *PEPs*) e as respostas das avaliações (envidas pelos *PDPs*).

5.2. Gerenciamento de Atributos

Os atributos da camada *IaaS* (Figura 3; *evento rsv*) refletem o consumo da máquina virtual como um todo (*e.g. CPU-v*). A camada *PaaS* fornece os atributos do usuário que está consumindo o serviço (*e.g. Atr-Usuário*; *evento ua*). Os dados armazenados no *PIP* são enviados por agentes (*AGM*) instanciados nas máquinas virtuais dos provedores (*evento tp*). O *PIP* da federação fornece as informações de consumo individual para o consumidor (*PIPC*; *evento st*) e para o *Broker* (*evento ar*) monitorar os *SLAs*.

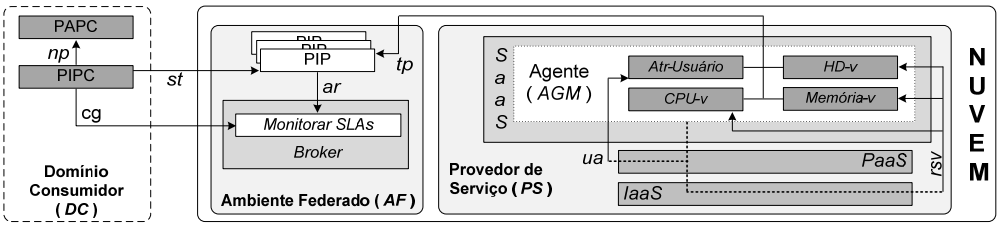


Figura 3. Gerenciamento de atributos de consumo

Com os atributos individuais (*evento st*) e o consumo consolidado (*evento cg*) é possível identificar quem está utilizando os serviços (*evento ua*) e a existência de recursos ociosos neste ambiente (*evento rsv*). Estes atributos são fornecidos para a administração de políticas do consumidor (*PAPC*; *evento np*) com intuito de otimizar a taxa de utilização dos recursos ou justificar a contratação de mais serviços.

5.3. Controle de Uso e Gerenciamento de Políticas

O administrador de políticas (*PAPC*) e o mecanismo de segurança (*MSDC*; Figura 4) transformam o *SLA* em regras de uso e credenciais de acesso (*evento re*). Utilizando as credenciais (*evento au*), os usuários acessam o Repositório de Interfaces (*RI*; *evento is*) do *Broker* e os serviços no provedor (*evento ac*). As políticas de uso são transferidas para a federação e armazenadas no *PAP* (*evento en*). Estas regras serão utilizadas para configurar a cota do usuário e para que um dos *PDP´s* pertencente ao *pool* de servidores da federação possa avaliar as solicitações de autorização (*eventos av, rp*). A avaliação utiliza os dados de consumo disponíveis no repositório de atributos (*PIP*; Seção 5.2).

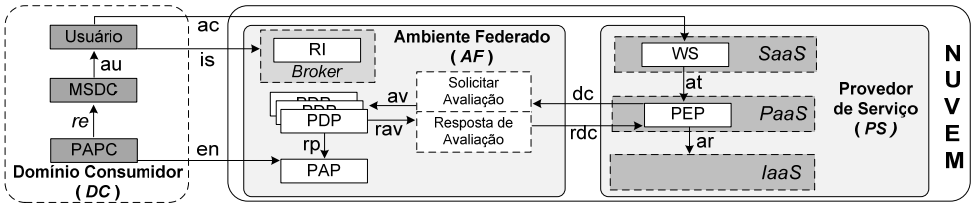


Figura 4. Gerenciamento de políticas de uso

Com a utilização do modelo *outsourcing* [Yavatkar *et al.*, 2010], o usuário precisa ser autorizado pelo *PDP* e ter o acesso liberado pelo *PEP* (camada *PaaS*; *evento at*) para poder consumir o recurso (camada *IaaS*; *evento ar*). Esta proposta libera o provedor e o consumidor da tarefa de implementar o monitor de referência $UCON_{ABC}$ e o gerenciamento de atributos de consumo (*i.e.* serviços fornecidos pela federação).

6. Protótipo e Testes de Avaliação

O protótipo utiliza os seguintes projetos e *APIs*: *Java* (*java.lang.management*); *SIGAR* (*hyperic.com/support/docs/sigar*); *JavaSysMon* (*jezhumble.github.com/javasysmon*); *Sun XACML* (*sunxacml.sourceforge.net*); *Apache TomCat* (*tomcat.apache.org*); módulo *Rampart* (*axis.apache.org/axis2/java/rampart*) integrado ao *Axis2* (*axis.apache.org*) para proteger as mensagens *SOAP* [W3C, 2007] (*i.e.* especificações *WS-Security* [OASIS, 2004] e *WS-Trust* [OASIS, 2007]); espaço de *tuplas River* (*river.apache.org).*

6.1. Testes e Avaliação

O serviço *web* [W3C, 2004] oferecido ao usuário simula um *e-commerce*. Nos testes, cada agente enviou uma *tupla* com *17* dados diferentes para o *PIP* (em torno de *2KB*). Os conteúdos enviados são: dados da máquina virtual que hospeda o serviço *web* (*e.g.* utilização do processador, memória); taxas referentes à *Java heap*; identificação da *thread* responsável por executar o serviço; tempo gasto pelo agente para coletar os atributos; tempo gasto pelo serviço para atender a solicitação do usuário (*overhead*).

As medidas foram obtidas executando-se *1000* interações e calculando a média entre estas. O coeficiente de variabilidade foi abaixo de *5%* em todos os casos. Os testes executados no espaço de *tuplas* (Tabela 1) visam identificar o número de entradas que este espaço consegue armazenar. Pode-se perceber que, na média, enquanto o tamanho da *tupla* dobra de valor, o número de entradas armazenadas reduz pela metade. Adicionalmente, o tempo gasto para armazenar cada *tupla* é alterado significativamente, resultando no melhor rendimento para entradas com tamanhos entre *4KB* e *16KB*.

Um teste similar foi executado no *PDP* para o processo de reavaliação de políticas (Tabela 2). Neste caso o *PDP* recupera a solicitação de avaliação do espaço de *tuplas*, (em torno de *2KB*), e a política do repositório (*PAP*). Na sequência, o *PDP* avalia a solicitação e envia o resultado para o espaço de *tuplas* correspondente. Este experimento mostrou a capacidade do *PDP* em atender as solicitações de reavaliação e o tempo gasto para executar a avaliação. De maneira semelhante, o tempo gasto pelo

PEP para escrever uma solicitação de avaliação no espaço de *tuplas* é similar ao tempo gasto pelos agentes (*AGM*) para escrever um atributo de consumo (*tupla* com *2KB*; Tabela 1).

As medidas apresentadas nas Tabelas 1 e 2 fornecem uma noção quanto ao "gatilho de elasticidade" da proposta, indicando que o número de servidores e serviços no *pool* deveria ser incrementado quando a demanda estiver próxima de causar a parada dos serviços. A abordagem de espaço de *tuplas*, juntamente com o esquema de avaliação distribuído (*i.e.* vários *PDPs* instanciados sob demanda) fornece elasticidade ao ambiente de avaliação $UCON_{ABC}$. O gerenciamento de atributos segue a mesma abordagem, fornecendo elasticidade ao sistema de contabilização.

Tabela 1: Armazenamento de *Tuplas*

Tamanho da tupla (incluindo 1KB de cabeçalho)	*Número de tuplas armazenadas antes do serviço recusar conexões*	*Tempo gasto para armazenar uma tupla (ms)*	*Throughput (KB/ms)*
2 KB	69864	2,54	55011
4 KB	45333	2,57	70557
8 KB	26651	3,01	70833
16 KB	14605	3,4	68729
32 KB	7670	4,94	49684
64 KB	3935	7,9	31878
128 KB	1492	13,44	14209
256 KB	1002	25,39	10102
512 KB	502	48,84	5262
1024 KB	250	98,88	2589
2048 KB	124	200,83	1264

Tabela 2: Reavaliação de Políticas

Tamanho da política recuperada do PAP	*Número de políticas reavaliadas paralelamente antes do serviço parar de responder*	*Tempo gasto para avaliar uma política (ms)*
4 KB	1690	0,129916
8 KB	1360	0,129123
16 KB	1080	0,128985
32 KB	640	0,130264
64 KB	470	0,131786
128 KB	310	0,133511
256 KB	220	0,14456
512 KB	130	0,144513
1024 KB	90	0,161233
2048 KB	70	0,186473

7. Conclusões

Esta seção será usada também para apresentar as principais *contribuições* da tese. O trabalho apresentou uma abordagem inovadora para a reavaliação de autorização contínua em controle de uso. O monitoramento constante de serviços e a reavaliação dos atributos de autorização permitem a identificação de disparidades entre autorizações e políticas. O esquema provê resiliência (*i.e.*, relaxamento das regras da política) para os atributos de autorização em algumas circunstâncias, sem perda para o consumidor (*e.g.*, violação de *SLA*). Quando o esquema de resiliência não é possível, o usuário estará em condição de exceção. Para este caso, o consumidor possui algumas alternativas para reparar a situação, o que não acontece nas abordagens tradicionais.

O serviço de contabilização proposto e a reavaliação contínua provê fina granularidade ao esquema de monitoramento e controle de acesso. A resiliência dos atributos de autorização (cotas) tornou o controle de acesso mais flexível. As violações em políticas de controle são monitoradas e tratadas no ambiente de gerenciamento da federação (*SLAs*) e do consumidor (condições de exceção). Este esquema permite o uso dos recursos sem ociosidade ou abuso de consumo dos serviços contratados.

A abordagem proposta mostrou que é possível executar o gerenciamento e a consolidação de atributos utilizando padrões abertos (*e.g.*, Serviços *Web*, Espaço de *Tuplas*). O esquema é adequado para o nível de acesso fornecido pela camada *IaaS*, não necessitando de mudanças no contexto atual. O gerenciamento é executado por serviços que trabalham de acordo com a demanda do consumidor. Sem estes serviços, não seria possível executar o controle fino de consumo, considerando que nenhum provedor de *IaaS* oferece serviços similares.

Resultados (publicações, patente e minicurso)

Marcon Jr, A. L., Santin, A. O., Stihler, M. e Bachtold, J. (2013). *A $UCON_{abc}$ Resilient Authorization Evaluation for Cloud Computing*. IEEE Transactions on Parallel and Distributed Systems, 11 April 2013. IEEE Computer Society Digital Library.

Marcon Jr, A. L., Santin, A. O., e Stihler, M. (2013). *Avaliação Resiliente de Autorização UCON$_{abc}$ para Computação em Nuvem*. XIII *SBSeg 2013*, pg. 16-29.

Marcon Jr, A. L., Santin, A. O., Stihler, M. *Método de Gerenciamento Elástico do Controle de Uso na Computação em Nuvem*. 2013. Patente: BR1020130267562. Data de depósito: 17/10/2013, INPI - Instituto Nacional da Propriedade Industrial.

Marcon Jr, A. L., Laureano, M., Santin, A. O. e Maziero, C. A. (2010). *Aspectos de Segurança e Privacidade em Ambientes de Computação em Nuvem*. X *SBSeg 2010*. *Anais de MiniCursos*, pg. 53-102.

Tese disponível em: *http://www.ppgia.pucpr.br/lib/exe/fetch.php?media=tese_-_arlindo_luis_marcon_junior.pdf*

Referências

Bertram S., Boniface M., Surridge M., Briscombe N. e Hall-May M. (2010). *On-Demand Dynamic Security for Risk-Based Secure Collaboration in Clouds*. 3rd CLOUD, pg. 518-525.

CSA (2011). *Security Guidance for Critical Areas of Focus in Cloud Computing v3.0*. Disponível: *cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf*. Aces. Mar. 2014.

Danwei C., Xiuli H. e Xunyi R., (2009). *Access Control of Cloud Service Based on UCON*. *1st CloudCom* 2009. LNCS. pg. 559-564.

Emeakaroha V. C., Netto M. A. S., Calheiros R. N., Brandic I., Buyya R. e Rose C. A. F. De. (2011). *Towards Autonomic Detection of SLA Violations in Cloud Infrastructures*. *Elsevier* FGCS, pg. 1-13.

Hayes B. (2008). *Cloud computing*. *Communications* ACM, vol. 51, no. 7, pg. 9-11.

Lim H. C., Babu S., Chase J. S. e Parekh S. S. (2009). *Automated Control in Cloud Computing: Challenges and Opportunities. 1st* ACDC, pg. 13-18.

Marcon Jr. A. L., Santin A. O., Lima Jr. L. A. de P., e Stihler M. (2009). *Policy Management Architecture Based on Provisioning Model and Authorization Certificates*. *ACM* SAC, pg. 1594-1598.

Marcon Jr, A. L., Laureano, M., Santin, A. O. e Maziero, C. A. (2010). *Aspectos de Segurança e Privacidade em Ambientes de Computação em Nuvem*. *Anais de MiniCursos do SBSeg 2010, SBC,* pg. 53-102.

Marcon Jr, A. L., Santin, A. O., Stihler, M. e Bachtold, J. (2013). *A UCON$_{abc}$ Resilient Authorization Evaluation for Cloud Computing*, *IEEE TPDS.*

Mell P. e Grance T. (2009). *The NIST Definition of Cloud Computing*. *Special Publication* 800-145. Acesso: Jan. 2014.

OASIS (2004). *Web Services Security SOAP Message Security 1.1*. Disponível em: *docs.oasis-open.org/wss/v1.1*. Acesso: Jan. 2014.

OASIS (2007). *WS-Trust 1.3*. Disponível em: *www.oasis-open.org/standards#wstrustv1.3*. Acesso: Jan. 2014.

Park J. e Sandhu R. (2004). *The UCON$_{ABC}$ Usage Control Model*. *ACM TISSEC*, vol. 7, no. 1, pg. 128-174.

Yavatkar R., Pendarakis D. e Guerin R. (2000). *A Framework for Policy-based Admission Control*, RFC 2753.

Tavizi T., Shajari M. e Dodangeh P., (2012). *A Usage Control Based Architecture for Cloud Environments*. *IEEE IPDPSW 2012*, *pg. 1534-1539*.

W3C (2004). *Web Services Architecture*. Available at: *www.w3.org/TR/ws-arch*. W3C (2007) *SOAP Version 1.2*. Disponível em: *www.w3.org/TR/soap*.

Zhang L. J. e Zhang J. (2009). *An Integrated Service Model Approach for Enabling SOA*. *IEEE* IT Pro. pg. 28-33.

TwinBFT: Tolerância a Faltas Bizantinas com Máquinas Virtuais Gêmeas

Fernando Dettoni[1], **Lau Cheuk Lung**[1]

[1]Departamento de Informática e Estatística - Universidade Federal de Santa Catarina - Brasil

{fdettoni, lau.lung}@inf.ufsc.br

Abstract. *Aiming to supply the need for security in information systems, a lot of approaches were proposed. Despite of being practical, most part of these approaches still lack in performance or have too strong requirements. We present an architecture and an algorithm for Byzantine fault-tolerant state machine replication using virtualization. Despite of existing for more than 30 years, virtualization is becoming more common, mainly because of cloud computing applications. Our algorithm explores the advantages of virtualization to reliably detect and tolerate faulty replicas, allowing the transformation of Byzantine faults into omission faults. Our approach reduces the total number of physical replicas from* $3f+1$*, in traditional approaches, to* $2f+1$*. Our approach is based on the concept of twin virtual machines, where there are a set of virtual machines in each physical host, each one acting as failure detector of its twin, by the validation of the messages sent.*

Resumo. *Visando suprir a necessidade de segurança no funcionamento de sistemas computacionais, diversas abordagens tolerantes a faltas bizantinas foram criadas. Apesar de terem fins práticos, a maior parte destas abordagens ainda apresenta um fraco desempenho ou requisitos que limitam seu uso em boa parte dos cenários reais. Nesta dissertação foi apresentada uma arquitetura e um algoritmo para replicação de máquina de estados tolerante a faltas bizantinas usando virtualização. A virtualização, apesar de existir há mais de 30 anos, vem se tornando cada vez mais comum, sendo muito utilizada em aplicações de computação em nuvens. São exploradas as vantagens fornecidas pela virtualização para detectar e tolerar réplicas faltosas, de forma a transformar ou reduzir faltas bizantinas em faltas de omissão. Com esta transformação, a abordagem apresentada é capaz de reduzir o número total de réplicas físicas necessárias de* $3f+1$*, em abordagens tradicionais, para* $2f+1$*. Foi criado o conceito de máquinas virtuais gêmeas. Neste contexto, um protótipo foi implementado e alguns experimentos foram realizados para obter medidas do desempenho da abordagem em uma execução prática.*

1. Introdução

1.1. Motivação

Cada vez mais, sistemas computacionais são usados em aplicações críticas e por isso necessitam operar corretamente apesar da presença de faltas. Estas faltas causam a parada total, como faltas de *crash*, ou arbitrárias (Bizantinas) [Lamport et al. 1982]. Uma das técnicas mais utilizadas para tolerância a faltas é a *replicação de máquina de estados* (RME) [Schneider 1990], que utiliza máquinas de estados determinísticas para oferecer um serviço

replicado (e.g. [Castro and Liskov 1999, Yin et al. 2003, Kotla et al. 2008]). Dentre estas, o algoritimo PBFT [Castro and Liskov 1999] é frequentemente considerado um pilar, pois foi o primeiro algoritmo com desempenho suficiente para muitas aplicações práticas.

O PBFT e vários outros algoritmos de replicação de máquinas de estados BFT têm um alto custo de implementação pois possuem uma resiliência de $n \geq 3f + 1$, ou seja, precisam de $n > 3f$ réplicas para tolerar f réplicas faltosas. Para diminuir esse custo, surgiram algumas abordagens que usam um *componente confiável* para limitar o comportamento das réplicas faltosas usando apenas $n \geq 2f + 1$ réplicas [Correia et al. 2004, Chun et al. 2007, Veronese et al. 2013]. Foram propostas também abordagens para executar apenas $f + 1$ réplicas, mantendo outras $2f$ *em espera*, ou seja, sem consumir tempo de CPU mas sendo ativadas em caso de falha [Distler et al. 2011, Wood et al. 2011].

Esta dissertação apresenta uma nova arquitetura, chamada TwinBFT, para replicação de máquina de estados tolerante a faltas Bizantinas eficiente baseada em virtualização. O objetivo é reduzir de $n \geq 3f + 1$ para $n \geq 2f + 1$ o número de máquinas físicas necessárias para tolerar f faltas. Além disso, o algoritmo apresentado reduz o número de passos de comunicação em funcionamento normal de 5 (do PBFT) para 3, sem a participação do cliente no acordo. Até onde sabemos, este é o primeiro algoritmo com este número de passos sem adotar uma abordagem especulativa.

A proposta consiste na utilização de conjuntos de *máquinas virtuais gêmeas* executando o mesmo serviço da aplicação em cada uma das $n \geq 2f + 1$ máquinas físicas do sistema. A ideia principal da proposta é utilizar cada máquina virtual como um detector de faltas para sua gêmea: ao enviar uma requisição para duas máquinas gêmeas, ambas devem fornecer a mesma resposta, caso contrário, ambas serão consideradas faltosas e sua resposta pode ser ignorada pelas outras réplicas.

1.2. Objetivos

1.2.1. Objetivo Geral

Demonstrar a possibilidade de melhoria na resiliência em algoritmos tolerante a faltas bizantinas a partir da redução do número de réplicas e da redução do número de passos de comunicação entre as réplicas em uma infraestrutura utilizando replicação de máquina de estados e vitualização.

1.2.2. Objetivos Específicos

A partir do objetivo geral, foi definido alguns objetivos específicos:

- Especificar um protocolo de replicação de máquina de estados tolerante a faltas bizantinas, reduzindo o número de réplicas necessárias de $3f + 1$ para $2f + 1$.
- Comprovar a validade do protocolo proposto a partir da utilização de provas de correção.
- Analisar o protocolo proposto com base no tempo necessário para execução de requisições e na quantidade de requisições executadas em uma unidade de tempo.

A Seção 2 apresenta uma descrição do nosso modelo de sistema e suposições. Uma explicação do algoritmo será apresentada na Seção 3. Após isso, a Seção 4 apresenta algumas avaliações da implementação do algoritmo e a Seção 5 resume as conclusões.

2. Modelo de Sistema

Uma representação da arquitetura do sistema é mostrada na Figura 1. O sistema é composto por um conjunto de *n* máquinas físicas (ou *hosts*) *H* = $\{h_1, h_2, \ldots, h_n\}$ sendo que $n \geq 2f + 1$ e f é o número máximo de máquinas físicas faltosas. Cada *host* da Figura 1 contém um gerenciador de máquinas virtuais (VMM ou hipervisor) com duas máquinas virtuais (vm_i, vm'_i),chamadas gêmeas, executando em cada uma, uma réplica ou processo, respectivamente p_i e p'_i. Ambos os processos $\{p_i, p'_i\}$ executam o mesmo serviço e se comunicam entre si para validar cada mensagem de saída antes de enviar para outros processos.

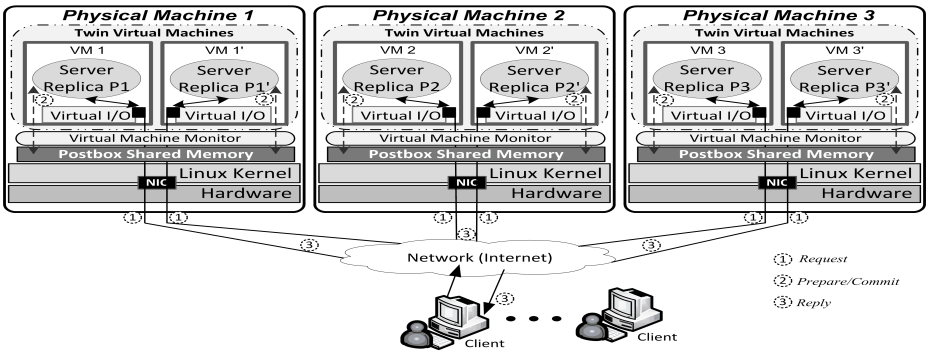


Figura 1. TwinBFT - Arquitetura com Máquinas Virtuais Gêmeas.

Supomos que até f máquinas virtuais podem falhar de forma Bizantina, ou arbitrária, mas apenas uma em cada máquina física. Quando uma máquina virtual falha arbitrariamente, o mecanismo de validação transforma essa falha numa omissão. Assim, supomos também que até f máquinas físicas podem falhar por paragem ou omitindo mensagens, acidentalmente ou devido à falha de uma das suas máquinas virtuais.

Nenhuma suposição é feita sobre o tempo necessário para o sistema computar uma mensagem. A comunicação entre diferentes VMs dentro de um mesmo *host* é feita por um espaço de memória compartilhada, chamada *postbox*. Os processos nas VMs, em diferentes hosts, se comunicam pela rede, apenas por troca de mensagens. Esta rede pode falhar ao entregar mensagens, entregar fora de ordem, atrasar, ou duplicar mensagens.

Cada *host* pode assumir dois diferentes papéis: (1) *host* primário, sendo responsável por definir a ordem em que as requisições dos clientes serão executadas; e (2) *host* backup, que executa as requisições seguindo a ordem proposta pelo primário. Dentro de um *host* primário, um processo pode assumir dois diferentes papéis: (1) líder, que é responsável por atribuir um número de sequência para cada requisição recebida; e (2) seguidor, que executa as requisições seguindo a ordem definida pelo líder. Todos os processos em *hosts* backups são considerados seguidores. O *host* primário h_j é definido por $j = v \bmod |S|$, sendo *v* a visão atual, conforme definido na próxima seção. O processo líder primário em um *host* é, por definição, p_j.

Foram utilizadas técnicas criptográficas para autenticar mensagens e garantir sua autenticidade. Cada par de processos compartilha entre si uma chave secreta usada para gerar um vetor de MACs (*Message Authentication Code*) [Tsudik 1992] com um MAC válido para cada processo.

3. Algoritmo TwinBFT

O algoritmo implementa uma replicação de máquina de estados no modelo de sistema que acabamos de apresentar. As réplicas se alternam seus papéis por uma sucessão de configurações chamadas de visão. Em cada visão, temos uma réplica primária que é responsável por definir a ordem das mensagens e encaminhar as requisições para todas as réplicas. Como mostrado por Schneider [Schneider 1990], a máquina de estados deve ser determinística e todas as réplicas precisam iniciar em um mesmo estado, caso contrário, a propriedade *safety* não pode ser garantida.

3.1. Propriedades

Sendo uma Replicação de Máquina de Estados, é necessário assegurar as seguintes propriedades para garantir a corretude do serviço: **Ordem Total** (*safety*) e **Terminação** (*liveness*)

O algoritmo proposto fornece tanto *safety* quanto *liveness*, supondo que não mais do que $f = \lfloor \frac{n-1}{2} \rfloor$ *hosts* são faltosos e, ao menos um processo *p* seja correto em cada *host* faltoso. Para garantir que as réplicas executarão as requisições na mesma ordem, todas as réplicas seguem a ordem definida pelo líder e esta ordem pode ser considerada correta desde que assinada por ambos os processos na réplica primária. Os algoritmos asseguram a corretude (ou *safety*) apesar do tempo levado para o processamento das requisições, porém uma certa sincronia é necessária para garantir as propriedades de progresso (i.e. *liveness*).

Como todas as réplicas seguem a ordem definida pelo líder primário, não é necessário um algoritmo de consenso pois pode-se confiar na ordem definida pela réplica primária, desde que as suposições prévias não sejam violadas. Isto ocorre porque quando o primário define a ordem, esta ordem apenas será seguida se ambos os processos no *host* primário tiverem acordo.

3.2. Operação normal do protocolo

Na Figura 2 é mostrado um diagrama com uma visão geral dos passos do protocolo. A configuração mostrada assume $f = 1$, sendo necessários três *hosts*, cada um com duas VMs. Cada par de VMs em um mesmo *host* se comunica através de um canal confiável FIFO chamado *postbox*. A *postbox* pode ser mais rápida do que a comunicação via rede, usando um espaço de memória compartilhado entre as VMs, fornecido pela VMM.

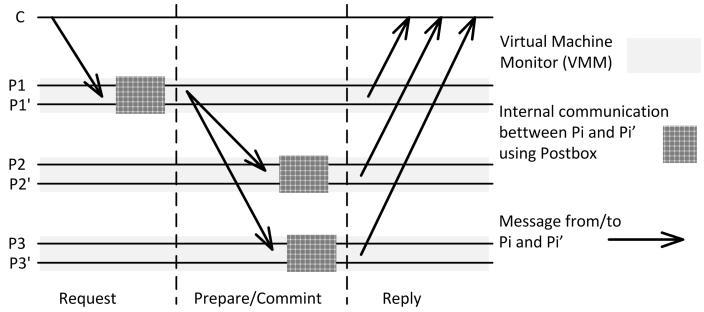


Figura 2. Passos do algoritmo em execução normal com $f = 1$.

1. Cliente envia a requisição para ambos os processos na réplica primária;
2. O líder primário p_i define um número de sequência e insere uma mensagem “ORDER"na *postbox*;

3. O seguidor primário p'_i lê a mensagem da *postbox*, pega o número de sequência e insere na postbox a mensagem "ORDER" contendo a requisição original e o número de sequência recebido;
4. Ambos os processos assinam a mensagem "ORDER" lida da *postbox* e enviam para todas as réplicas;
5. Assim que cada processo nas réplicas recebe a mensagem "ORDER", executa a operação e insere na *postbox* uma mensagem "REPLY" assinada;
6. Quando um processo lê da *postbox* uma mensagem "REPLY", compara com a mensagem gerada localmente e, se todos os argumentos forem idênticos, adiciona sua assinatura e envia a resposta para o cliente;
7. Se o cliente receber ao menos $f + 1$ respostas corretamente assinadas de diferentes réplicas, aceita a resposta.

Para cada mensagem "ORDER" recebida, as réplicas consideram válida caso as seguintes condições estejam cumpridas:

- A mensagem é corretamente assinada;
- A visão na mensagem é a visão atual;
- Não aceitou outra mensagem "ORDER" com o mesmo número de sequência para outra requisição;
- A número de sequência está entre um valor mínimo *h* e máximo *H* de possíveis números de sequência.

Quando o cliente recebe uma mensagem "REPLY", aceita como válida se as seguintes condições forem verdadeiras:

- Está assinada por ambos os processos $\{p_i, p'_i\}$ no *host* remetente.
- Ainda não aceitou uma mensagem válida remetida pela gêmea do *host* remetente.

O cliente aguarda até ter recebido ao menos $f + 1$ mensagens válidas das réplicas para aceitar o resultado. Se estas mensagens não forem recebidas em um determinado tempo, envia a requisição para todas as réplicas, podendo ocorrer uma troca de visão por suspeita do primário.

3.3. Coleta de Lixo

Para prevenir que ocorra um estouro na memória, é necessário um mecanismo que descarte as mensagens antigas armazenadas no *buffer*, isto é, que efetue uma coleta de lixo (i.e. *garbage collection*). Para isso, o algoritmo gera periodicamente um *checkpoint* e sincroniza o estado atual entre todas as réplicas.

3.4. Protocolo de Troca de Visão

A principal função do protocolo de troca de visão é manter o serviço progredindo mesmo na presença de um primário faltoso. Se o primário é faltoso, as réplicas backup nunca receberão uma mensagem "ORDER" válida e, portanto, devem definir um novo primário. Se um cliente não receber respostas suficientes em um tempo hábil, envia a requisição para todos os processos do sistema e inicia o processo de troca de visão.

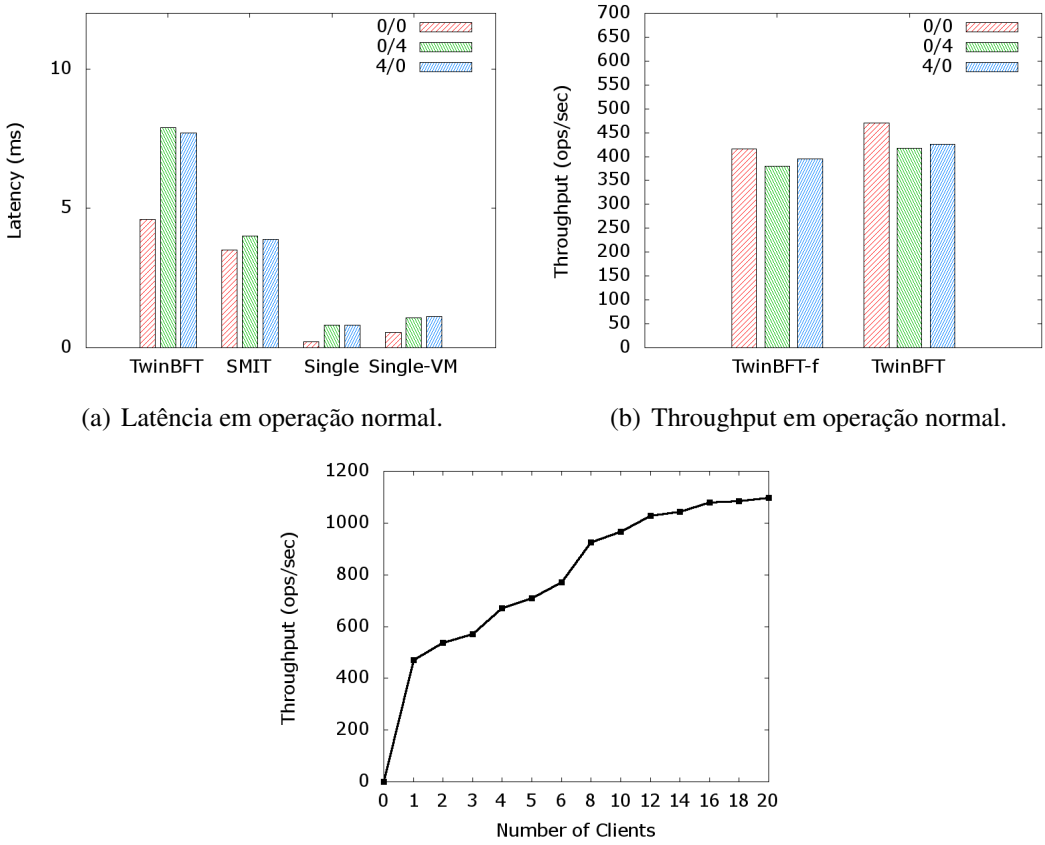
4. Implementação e Resultados

Para avaliar a performance da abordagem, foi escolhido um método experimental. O algoritmo foi implementado em Java, de acordo com as especificações da versão 1.6. Os canais

de comunicação foram feitos pela utilização de *channels* do Java NIO, usando TCP com MACs (Códigos de Autenticação de Mensagem).

Executamos nossos experimentos em três servidores Intel®Core™i7 3.8Ghz com Debian 7.0 "wheezy" (Kernel 3.2.0 x86-64) e VMM Xen Hypervisor 4.1.3. Cada máquina virtual foi configurada com 2GB de memória e duas CPUs virtuais, equipadas com SUN's JDK 1.6.0_29.

Como medida de avaliação, foi adotado latência e *throughput*, por serem largamente utilizadas neste tipo de avaliação e porque permitem uma verificação simplificada da eficiência do sistema [Jain 1991]. Os resultados foram obtidos através de *microbenchmarks* em diferentes condições de carga. A latência foi obtida a partir de algumas requisições com um único cliente enviando uma requisição por vez, e o *throughput* medindo quantas requisições o sistema consegue responder em uma unidade de tempo. O sistema foi avaliado a partir de *microbenchmarks* para que o custo fosse avaliado sem a influência do serviço. Para estes *microbenchmarks*, foi utilizado um serviço *stateless* com uma operação nula, variando o tamanho das requisições e respostas entre 0KB e 4KB.



(a) Latência em operação normal.

(b) Throughput em operação normal.

(c) Throughput com múltiplos clientes.

Figura 3. Desempenho verificado para o *TwinBFT* em operação normal.

A fim de avaliar o desempenho da solução proposta, o algoritmo foi executado em condições, em que foram enviadas 10.000 requisições de um único cliente, em três diferentes cargas: 0/0, 0/4 e 4/0. Eles representam, respectivamente, uma requisição e resposta nula, uma requisição nula e uma resposta de 4KB, e uma requisição de 4KB e resposta nula. Todos os tempos foram medidos pelo cliente, a partir da leitura de seu

relógio local antes do envio da requisição e após o recebimento de uma resposta válida.

Na Figura 3(a), são mostradas as diferentes latências em cada carga. Para obter a latência, 10.000 requisições foram enviadas e executadas sequencialmente, de forma individual, sendo a latência o tempo médio destas requisições. A abordagem é comparada com o algoritmo SMIT [Stumm et al. 2010], que contém certas semelhanças pela utilização de máquinas virtuais e memória compartilhada, porém não suporta faltas de *crash*. *Single* se refere a uma implementação do serviço centralizado, sem máquinas virtuais enquanto *Single-VM* representa uma execução centralizada dentro do isolamento de uma máquina virtual. É esperado que a abordagem proposta apresente uma avaliação pior do que versões centralizadas pois estas não são tolerantes a faltas.

O *throughput*, como mostrado na Figura 3(b) foi calculado baseado no tempo total para execução das 10.000 requisições enviadas simultaneamente para o serviço. No primeiro grupo, são mostradas as medidas para o serviço em seu caso normal, sem falhas. Em TwinBFT-f mostramos o *throughput* do algoritmo em caso de faltas, sendo que em 1% das requisições a réplica primária se mostra faltosa, gerando uma troca de visão para manter a consistência. Para uma comparação do desempenho em diferentes cargas do algoritmo, a Figura 3(c) apresenta o *throughput* quando temos mais de um cliente fazendo requisições simultaneamente. O número de requisições por segundo se eleva com o aumento de clientes simultâneos até se estabilizar em torno de 1000 operações por segundo quando passa a ficar mais limitado pela capacidade das réplicas do que pela capacidade dos clientes.

5. Conclusões

A partir da exploração de algumas técnicas de virtualização, foi possível a proposta de um algoritmo BFT alternativo. Foi mostrado que é possível implementar um algoritmo RME confiável com $2f + 1$ réplicas físicas em um ambiente assíncrono. Apesar de necessitar de um canal de comunicação confiável para a comunicação entre as máquinas virtuais, acreditamos que a virtualização é vastamente disponível atualmente e pode fornecer um isolamento entre as réplicas e o mundo exterior. Além do mais, foi possível reduzir também o número de passos de comunicação, reduzindo o custo desta comunicação.

O desenvolvimento deste trabalho resultou na publicação dos seguintes artigos em eventos e periódicos da área:

- *"Byzantine Fault-Tolerant State Machine Replication with Twin Virtual Machines"* - 18th IEEE Symposium on Computers and Communications - IEEE ISCC 2013 [Dettoni et al. 2013a].
- *"Using Virtualization Technology for Fault-Tolerant Replication in LAN"* - 8th International Conference on Dependability and Complex Systems - publicado no periódico "Advances in Intelligent and Soft Computing" - 2013 [Dettoni et al. 2013c].
- *"Replicação por Máquina de Estados Tolerante a Faltas Bizantinas usando Máquinas Virtuais Gêmeas"* - XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2013 [Dettoni et al. 2013b].

O trabalho completo está disponível para acesso no endereço http://tede.ufsc.br/teses/PGCC0977-D.pdf [DETTONI 2013].

Referências

Castro, M. and Liskov, B. (1999). Practical Byzantine fault tolerance. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, pages 173–186.

Chun, B.-G., Maniatis, P., Shenker, S., and Kubiatowicz, J. (2007). Attested append-only memory: making adversaries stick to their word. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles*, pages 189–204.

Correia, M., Neves, N. F., and Verissimo, P. (2004). How to tolerate half less one Byzantine nodes in practical distributed systems. In *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems*, pages 174–183.

Dettoni, F., Lung, L. C., Correia, M., and Luiz, A. F. (2013a). Byzantine fault-tolerant state machine replication with twin virtual machines. In *18th IEEE Symposium on Computers and Communications (IEEE ISCC 2013)*.

Dettoni, F., Lung, L. C., Correia, M., and Luiz, A. F. (2013b). Replicação por máquina de estados tolerante a faltas bizantinas usando máquinas virtuais gêmeas. In *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Brasília, DF, Brasil.

Dettoni, F., Lung, L. C., and Luiz, A. F. (2013c). Using virtualization technology for fault-tolerant replication in lan. In *8th International Conference on Dependability and Complex Systems*.

DETTONI, F. A. (2013). Twinbft: tolerância a faltas bizantinas com máquinas virtuais gêmeas. Dissertação (mestrado), Universidade Federal de Santa Catarina.

Distler, T., Popov, I., Schröder-Preikschat, W., Reiser, H. P., and Kapitza, R. (2011). SPARE: Replicas on hold. In *Proceedings of the 18th Network and Distributed System Security Symposium*, pages 407–420.

Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons.

Kotla, R., Clement, A., Wong, E., Alvisi, L., and Dahlin, M. (2008). Zyzzyva: speculative Byzantine fault tolerance. *Commun. ACM*, 51:86–95.

Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401.

Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: a tutorial. *ACM Comput. Surv.*, 22(4):299–319.

Stumm, V., Lung, L. C., Correia, M., da Silva Fraga, J., and Lau, J. (2010). Intrusion tolerant services through virtualization: A shared memory approach. In *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, pages 768 –774.

Tsudik, G. (1992). Message authentication with one-way hash functions. *SIGCOMM Comput. Commun. Rev.*, 22(5):29–38.

Veronese, G. S., Correia, M., Bessani, A. N., C., L., and Verissimo, P. (2013). Efficient Byzantine fault tolerance. *IEEE Transactions on Computers*, 62(1):16–30.

Wood, T., Singh, R., Venkataramani, A., Shenoy, P., and Cecchet, E. (2011). ZZ and the art of practical BFT execution. In *Proceedings of the 6th ACM SIGOPS/EuroSys European Systems Conference*, pages 123–138.

Yin, J., Martin, J.-P., Venkataramani, A., Alvisi, L., and Dahlin, M. (2003). Separating agreement from execution for Byzantine fault tolerant services. *SIGOPS Oper. Syst. Rev.*, 37:253–267.

Emparelhamentos e reticulados: estado-da-arte em algoritmos e parâmetros para as famílias mais flexíveis de sistemas criptográficos

Jefferson E. Ricardini

[1]Escola Politécnica, University of São Paulo, Brazil.
Orientador: Paulo S. L. M. Barreto

{jricardini,pbarreto}@larc.usp.br

Abstract. *Public key cryptography is an area of knowledge undergoing intense research at present. Some cryptographic primitive families tend to be extremely prolific in terms of flexibility, efficiency and security, among then we find the* pairings *and the* lattices*. Because of their similar functionalities and their rare versatility within the whole area of cryptography despite their completely disparate nature, some authors proposed to call lattices "the new pairings," according to the chronological order by which they began to attract more vivid research interest. In this scenario, a comparative study between them is of reasonable interest, in particular on the advantages and disadvantages that the state of the art reveals about the efficiency of each one. The thesis described herein addresses this comparative study, and also contributes efficient pairing implementation techniques, new parameters for building compact lattices and an innovative technique to instantiate lattices in practice.*

Resumo. *A criptografia de chave pública é uma área do conhecimento sujeita a intensa atividade de pesquisa. Algumas famílias de primitivas criptográficas mostram-se extremamente versáteis em termos de flexibilidade, eficiência e segurança, entre estas estão os* emparelhamentos *e os* reticulados*. Por possuírem semelhanças em suas funcionalidades, e exibirem uma versatilidade rara na a área de criptografia, a despeito de suas naturezas completamente díspares, alguns autores propuseram chamar os reticulados de "os novos emparelhamentos", conforme a ordem cronológica em que elas passaram a atrair interesse mais vívido de pesquisa. Neste cenário, um estudo comparativo entre elas é de razoável interesse, em particular sobre vantagens e desvantagens que o estado da arte revela sobre a eficiência de cada uma delas. A dissertação aqui descrita contempla esse estudo, e contribui técnicas de implementação eficiente de emparelhamentos, novos parâmetros para a construção de reticulados compactos e uma técnica inovadora para instanciar reticulados na prática.*

1. Introdução e Motivação

Emparelhamentos constituem uma primitiva de criptografia tradicional e tecnologicamente madura, pois possui implementações extremamente otimizadas em diversos plataformas [Aranha et al. 2011, Pereira et al. 2011]. A criptografia baseada em emparelhamentos atrai muita atenção graças a sua flexibilidade e versatilidade, que permitem

a elaboração de diversas aplicações e protocolos, desde assinaturas digitais e encriptação comuns até aplicações mais complexas, como criptossistemas baseados em identidade [Sakai et al. 2000], assinaturas em anéis [Chen et al. 2006, Zhang and Kim 2002], assinaturas cegas [Zhang and Kim 2002] e protocolos de acordo de chaves autenticados [Chen et al. 2007].

Por outro lado, recentemente os reticulados têm se tornado, cada vez mais, alvo de intensa pesquisa. Devido a sua grande flexibilidade e versatilidade em aplicações criptográficas, alguns autores se motivaram a chama-los de "os novos emparelhamentos". No entanto, sob a perspectiva de maturidade tecnológica, ainda há muita incerteza, pois até o momento em que a dissertação foi escrita não foram encontradas implementações tão otimizadas e eficientes como em emparelhamentos. Em geral, criptografia baseada em reticulados ainda necessita de chaves grandes e possui um *overhead* de assinaturas e criptogramas também grandes em comparação com o que é encontrado em criptografia de curvas elípticas, tipicamente usadas em emparelhamentos.

Outro problema refere-se a técnicas de amostragem de Gaussianas discretas, que é o gargalo de processamento em diversas aplicações. Esse tipo de amostragem é necessária nas operações básicas dos protocolos, seja geração de chaves, encriptação ou assinatura. em geral, para a garantia de uma prova formal de segurança ou para o próprio funcionamento correto do protocolo.

Tendo em vista o que foi citado acima, é um interesse de pesquisa o confronto entre essas técnicas a fim de verificar até que ponto os reticulados podem ser comparados aos emparelhamentos, principalmente no que tange à eficiência dos algoritmos, bem como ao espaço necessário para chaves, assinaturas e criptogramas.

Na dissertação aqui descrita foram analisados os estados da arte de ambas as primitivas, com o intuito de confrontar os aspectos de flexibilidade e maturidade tecnológica. Também foi proposta uma alternativa de parâmetros para reticulados, evitando a utilização de anéis circulantes e polinômios ciclotômicos, muito usados em reticulados hoje em dia. Foi proposta ainda uma estratégia mais eficiente para a amostragem de gaussianas discretas, necessárias em muitos protocolos baseados em reticulados.

2. Motivaçao e objetivos

Por se tratar de uma primitiva já estabelecida e possuir muitas aplicações e implementações eficientes e otimizadas para diversas plataformas, a criptografia baseada em emparelhamentos já é considerada tradicional. No entanto, ela não atende às métricas exigidas em certos cenários com dispositivos de capacidade limitada de recursos como processamento, memória ou largura de banda, como é o caso de redes de sensores sem e dispositivos normalmente utilizados em aplicações do tipo "Internet das Coisas" [Margi et al. 2010].

Em contrapartida, a criptografia baseada em reticulados ainda é uma área de pesquisa que tem muito a ser explorada, e de fato para chegar a um nível de maturidade comparável ao dos emparelhamentos ainda são necessários muitos avanços. Os reticulados tem a vantagem de possuírem operações básicas muito simples e necessitarem de corpos finitos muito menores do que os encontrados em primitivas que envolvem emparelhamentos. Essa característica e uma vantagem importante, pois a simplicidades das operações necessária torna essa família ideal para aplicações do tipo "Internet das Coisas".

O objetivo principal dessa dissertação foi confrontar as primitivas criptográficas baseadas em emparelhamentos com as baseadas em reticulados, tanto de maneira qualitativa (versatilidade para a construção de protocolos) quanto quantitativa (últimos avanços e o estado da arte da implementação eficiente).

Em complemento, este trabalho contemplou a apresentação de alternativas para alguns dos aspectos mais essenciais dessas primitivas. Especificamente, abordam-se aqui, por um lado, a construção de algoritmos mais eficientes para o cálculo de emparelhamentos afins (área ainda escassamente explorada na literatura), e por outro, a proposta de uma nova álgebra (igualmente eficiente, mas livre de patentes) para representar reticulados e técnicas efetivas para a amostragem de variáveis normais em reticulados, necessária em certas operações de muitos protocolos em criptografia baseada em reticulados.

3. Contribuições

A dissertação contemplou as seguintes contribuições.

- Avaliação sinóptica dos aspectos algorítmicos de duas famílias de primitivas criptográficas de naturezas diferentes, uma já bem estabelecida e tecnologicamente madura (emparelhamentos bilineares) e outra ainda considerada experimental e vista como alternativa (reticulados), mas ambas com um potencial imenso atestado pela literatura abundante acerca de cada uma delas, a ponto de alguns pesquisadores afirmarem sugestivamente que os reticulados seriam os "novos emparelhamentos".
- Proposta de uma nova parametrização de criptossistemas baseados em reticulados, como alternativa aos reticulados ideais utilizados atualmente (anéis polinomiais circulantes, negacíclicos e ciclotômicos em geral). Especificamente, avaliam-se aqui as denominadas álgebras discretas de Rojo, baseada nas álgebras contínuas propostas por Oscar Rojo [Rojo 2008], e explora-se como adaptar essas álgebras ao contexto de anéis discretos e finitos, necessários para aplicações criptográficas com reticulados. Tal parametrização, além de evitar as patentes referentes aos parâmetros mais comuns adotados em criptossistemas baseados em reticulados (e.g. NTRU [Hoffstein et al. 1998]) , revela-se tão compacta e quase tão eficiente quanto as outras alternativas.
- Otimização nas estratégias utilizadas para amostragem de Gaussianas discretas. Propõem-se a utilização da transformada rápida de Walsh-Hadamard (FWHT) para refinar uma amostragem normal de n variáveis criptográficas simultaneamente (conforme o cenário comum de reticulados com dimensão n), com custo total amortizado próximo da complexidade ideal, a saber, $O(\lg n)$ por variável.
- Como bônus e fruto de pesquisa realizada em parceria com Diego F. Aranha (UnB) e Patrick Longa (MSR), aperfeiçoou-se neste trabalho o estado da arte no cálculo de emparelhamentos, com uma implementação mais eficiente (recordista de desempenho na literatura) em coordenadas afins e projetivas [Aranha et al. 2013].

4. Resultados

Além das conclusões a cerca da avaliação sinóptica, dos aspectos algorítmicos das famílias criptográficas estudadas durante o mestrado, e discutidas na seção 5 deste documento, o trabalho realizado teve resultados independentes em cada uma das famílias estudas. Estes resultados são apresentados a seguir.

Emparelhamentos

Foi feita uma implementação de referência em linguagem C, com pontos críticos de eficiência implementados em *assembly*, usando o compilador GCC versão 4.7.0. Essa implementação é agora parte integral da biblioteca RELIC [Aranha and Gouvêa 2013]. Os experimentos para medição de eficiência foram feitos em um conjunto de plataformas compatíveis com Intel de 64 bits: processadores mais antigos Nehalem Core i5 540M 2.53GHz e AMD Phenom II 3.0 GHz, e processadores mais modernos Sandy Bridge Xeon E31270 3.4GHz e Ivy Bridge Core i5 3570 3.4GHz, incluindo um processador mais recente Haswell Core i7 4750 HQ 2.0GHz. Todas as máquinas tiveram seu sistema de *overclock* automático desativado para aumentar a confiabilidade dos resultados.

A tabela 1 apresenta os tempos divididos em laço de Miller e exponenciação final e expressos em quilociclos de *clock*, e tomados como a média de 10^4 repetições de cada operação. As plataformas são Intel Nehalem (N) AMD Phenom II (P II), Sandy Bridge (SB), Ivy Bridge (IB), Haswell sem e com o uso da instrução `mulx` (H, H+`mulx`).

Tabela 1. Implementações em plataformas de 64 bits (tempos em quilociclos).

Operação	N	P II	SB	IB	H	H+`mulx`
Laço de Miller afim	1680	1341	1365	1315	1259	1212
Laço de Miller proj.	1170	862	856	798	721	704
Exponenciação final	745	557	572	537	492	473
Emp. afim	2425	1898	1937	1852	1751	1685
Emp. proj.	1915	1419	1428	1335	1213	1177

Reticulados

Para aferir o efeito do método de amostragem via FWHT e da adoção de álgebras discretas de Rojo, implementaram-se em linguagem C as seguintes primitivas:

1. Amostragem pelo método da razão [Kinderman and Monahan 1977], pelo método polar [Marsaglia and Bray 1964] e pelo método FWHT modificado conforme proposto na dissertação;
2. Aritmética em anéis ideais negacíclicos $\mathbb{Z}[x]/(x^n + 1)$ e em álgebras discretas de Rojo;
3. Criptossistema Lindner-Peikert [Lindner and Peikert 2011], considerado na literatura um dos mais eficientes para cifração com reticulados;
4. Benchmarks para todas as combinações de método de amostragem e criptossistemas para níveis realísticos de segurança.

As propriedades dos esquemas implementados estão listadas na tabela 2 (tamanhos medidos em bits).

Os resultados de tempo obtidos são sumarizados na tabela 3 (tempos em μs).

Tabela 2. Propriedades de criptossistemas baseados em reticulados

sistema	n	segurança	$\|pk\|$	$\|ct\|$
LP	512	2^{97}	15380	30760
LP	1024	2^{247}	33273	66546

Tabela 3. Desempenho de criptossistemas baseados em reticulados

sistema	n	álgebra	amostragem	GeraChaves	Encripta	Decripta
LP	512	negacíclica	razão	174.6	256.5	25.5
LP	512	negacíclica	polar	124.4	216.1	26.1
LP	512	negacíclica	FWHT	71.6	132.8	25.6
LP	512	Rojo	razão	208.8	341.9	46.3
LP	512	Rojo	polar	171.2	305.5	46.6
LP	512	Rojo	FWHT	118.4	222.6	46.4
LP	1024	negacíclica	razão	354.0	571.5	68.1
LP	1024	negacíclica	polar	281.2	475.8	65.9
LP	1024	negacíclica	FWHT	174.5	324.1	64.4
LP	1024	Rojo	razão	468.5	783.0	119.1
LP	1024	Rojo	polar	387.1	714.8	120.3
LP	1024	Rojo	FWHT	288.7	546.0	118.0

Os resultados, obtidos num PC Windows 7, Intel i5 2.5 GHz, mostram que, para um dado método de amostragem de variáveis normais, o desempenho típico de criptossistemas sobre álgebras discretas de Rojo é, conforme o método adotado de amostragem normal, entre 20% e 65% mais lento para geração de chaves, entre 35% a 70% mais lento para encriptação, e entre 75% e 85% mais lento para decriptação que esses mesmos criptossistemas sobre anéis negacíclicos.

O comportamento verificado não surpreende, já que o método adotado para a implementação de aritmética de Rojo recorre diretamente à FFT sobre uma álgebra *duas vezes maior*. Este resultado é melhor descrito na dissertação. Com relação ao tamanho das chaves, a álgebra de Rojo descrita na dissertação se mostrou tão compacta quanto ao estado da arte na literatura, possuindo os mesmo tamanhos de chaves das parametrizações mais eficientes.

5. Discussões

Para fins de uma comparação breve e abrangente de emparelhamentos e reticulados, foi feita uma comparação sinóptica dessas primitivas em termos de flexibilidade, eficiência e segurança. As conclusões gerais descritas na dissertação são relatadas abaixo.

Flexibilidade

As primitivas analisadas são extremamente flexíveis, e ocorrem em uma série de aplicações e protocolos. Ambas as primitivas contemplam tanto aplicações elementares de criptografia de chave pública (encriptação e assinaturas digitais comuns) quanto esquemas mais avançados.

No primeiro aspecto (aplicações elementares), os emparelhamentos e os reticulados podem ser considerados igualmente flexíveis, pois as aplicações e protocolos que podem ser feitos em uma primitiva também podem ser feitos com a outra.

No segundo aspecto, embora existam protocolos implementáveis com ambas as primitivas (por exemplo, encriptação baseada em identidade e cifrassinaturas), outros protocolos são implementáveis completamente somente com uma das primitivas (por exemplo, assinaturas cegas com curvas elípticas e emparelhamentos, ou encriptação homomórfica com reticulados). Com isso, embora imensuráveis, ainda se podem considerar as duas primitivas como "equivalentes" em termos de flexibilidade.

Embora, os protocolos avançados baseados em emparelhamentos são diretamente utilizáveis em aplicações práticas, enquanto que muitos dos protocolos avançados em reticulados são apenas provas de conceito na literatura existente, não sendo remotamente eficientes para serem usados na prática.

Eficiência

Sob o ponto de vista da eficiência, verificou-se que os emparelhamentos admitem uma gama de otimizações muito maior do que a criptografia baseada em reticulados, abrangendo as mais diversas plataformas computacionais. Nesse sentido, há uma grande quantidade de trabalhos recentes sobre emparelhamentos que se dedicam à implementação otimizada para uma plataforma específica. Essa mudança de foco é uma evidência de que otimizações e melhorias teóricas mais gerais já são mais difíceis de se encontrar, ou seja, que a complexidade teórica do cálculo de emparelhamentos é um assunto bem compreendido.

Por outro lado, nos trabalhos sobre reticulados ainda são encontradas otimizações mais gerais ou assintóticas. Em geral, essas parametrizações visam reduzir o tamanho das chaves e o *overhead* em mensagens encriptadas e assinaturas digitais, e reduzir a complexidade teórica de processamento.

A criptografia baseada em emparelhamentos é, portanto, substancialmente mais madura do que as primitivas baseadas em reticulados. Por outro lado, essa menor maturidade dos reticulados (além da intensa pesquisa e interesse crescente) sugere oportunidades inexploradas de otimização que aparentemente já foram esgotadas com emparelhamentos.

Segurança

As diferenças entre emparelhamentos e reticulados continuam no aspecto de segurança. Os emparelhamentos são baseados em problemas considerados tradicionais, bem explorados e bem estabelecidos em criptografia. Ataques bem sucedidos mais recentes existem, mas podem ser evitados desde que a escolha dos parâmetros seja adequada, devendo-se dar atenção desde a escolha dos corpos finitos até a escolha de curvas adequadas.

Por outro lado, os reticulados chamam atenção por haver reduções de segurança do caso médio em certos reticulados para o pior caso em reticulados relacionados. Essa propriedade, se for bem aplicada, oferece parâmetros em que essencialmente qualquer escolha aleatória de chaves alcança o nível desejado de segurança.

6. Conclusões

O trabalho aqui descrito confrontou duas primitivas criptográficas diferentes. A criptografia baseada em emparelhamentos e a criptografia baseada em reticulados. Esse confronto se deu tanto nos aspectos de flexibilidade e versatilidade para a construção de protocolos, quanto nos últimos avanços e o estado da arte da implementação eficiente.

Após esse confronto, chegou-se a conclusão que, considerando os aspectos de flexibilidade, eficiência e segurança, verificou-se que a proposição "os reticulados são os novos emparelhamentos" faz sentido. No entanto, os reticulados encontram-se talvez no mesmo nível de investigação que os emparelhamentos em seus primórdios, oferecendo grande potencial de evolução e exploração.

Além dessa comparação, houve também contribuições individuais em cada uma das áreas, mais especificamente, propôs-se uma nova álgebra, igualmente eficiente na métrica de tamanho de chaves, mas livre de patentes, para representar reticulados e técnicas efetivas para a amostragem de variáveis normais em reticulados. Como bônus e fruto de pesquisa realizada em colaboração com Diego F. Aranha (UnB) e Patrick Longa (MSR), obteve-se uma implementação recordista de desempenho em coordenadas afins, área ainda pouco explorada na literatura. Este trabalho foi publicado na conferência *Selected Areas in Cryptography* (SAC 2013)[Aranha et al. 2013].

Finalmente, em decorrência da pesquisa relatada dissertação aqui descrita, pode-se elencar como oportunidades de extensões e pesquisas futuras: a procura de mais álgebras alternativas às utilizadas atualmente e à álgebra discreta de Rojo sugerida nesses trabalho; uma análise formal da qualidade da amostragem normal com a FWHT; técnicas de implementação dessa amostragem para obter desempenho independente dos dados amostrados (importante para evitar ataques de canal secundário); e uma versão da transformada discreta de cossenos que dispense a duplicação implícita ou explícita do tamanho dos elementos da álgebra durante o cálculo da transformada.

A dissertação está disponível em `http://www.larc.usp.br/~joliveira/dissertacao.pdf` e em breve no banco de teses e dissertações da USP (`http://www.teses.usp.br/index.php`).

Referências

Aranha, D. F., Barreto, P. S. L. M., and Longa, P.and Ricardn, J. E. (2013). The realm of the pairings. In *Selected Areas in Cryptography – SAC 2013*, Lecture Notes in Computer Science, Vancouver, Canada. Springer. To appear.

Aranha, D. F. and Gouvêa, C. P. L. (2013). RELIC is an Efficient LIbrary for Cryptography. `http://code.google.com/p/relic-toolkit/`.

Aranha, D. F., Karabina, K., Longa, P., Gebotys, C. H., and López, J. (2011). Faster explicit formulas for computing pairings over ordinary curves. In *Advances in Cryptology – Eurocrypt 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 48–68, Tallinn, Estonia. Springer.

Chen, L., Cheng, Z., and Smart, N. P. (2007). Identity-based key agreement protocols from pairings. *International Journal of Information Security*, 6(4):213–241.

Chen, X., Zhang, F., and Kim, K. (2006). New ID-based group signature from pairings. *Journal of Electronics (China)*, 23(6):892–900.

Hoffstein, J., Pipher, J., and Silverman, J. (1998). NTRU: A ring-based public key cryptosystem. In Buhler, J. P., editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, Oregon, USA.

Kinderman, A. J. and Monahan, J. F. (1977). Computer generation of random variables using the ratio of uniform deviates. *ACM Transactions on Mathematical Software*, 3(3):257–260.

Lindner, R. and Peikert, C. (2011). Better key sizes (and attacks) for LWE-based encryption. In *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339, San Francisco, CA, USA. Springer.

Margi, C. B., de Oliveira, B. T., de Sousa, G. T., Jr., M. A. S., Barreto, P. S. L. M., Carvalho, T. C. M. B., Näslund, M., and Gold, R. (2010). Impact of operating systems on wireless sensor networks (security) applications and testbeds. In *ICCCN*, pages 1–6, Zurich, Switzerland. IEEE.

Marsaglia, G. and Bray, T. A. (1964). A convenient method for generating normal variables. *SIAM Review*, 6:260–264.

Pereira, G. C. C. F., Simplício, Jr., M. A., Naehrig, M., and Barreto, P. S. L. M. (2011). A family of implementation-friendly BN elliptic curves. *Journal of Systems and Software*, 84(8):1319–1326.

Rojo, O. (2008). A new algebra of Toeplitz-plus-Hankel matrices and applications. *Computers and Mathematics with Applications*, 55(12):2856 – 2869.

Sakai, R., Ohgishi, K., and Kasahara, M. (2000). Cryptosystems based on pairing. In *Symposium on Cryptography and Information Security – SCIS 2000*, Okinawa, Japan. Springer.

Zhang, F. and Kim, K. (2002). ID-based blind signature and ring signature from pairings. In Zheng, Y., editor, *Advances in Cryptology – Asiacrypt 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 533–547. Springer, Queenstown, New Zealand.

Illumination Inconsistency Sleuthing for Exposing Fauxtography and Uncovering Composition Telltales in Digital Images

Tiago Carvalho[1], **Hélio Pedrini**[1] **and Anderson Rocha**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brazil

{tjose, helio, anderson.rocha}@ic.unicamp.br

Abstract. *Once taken as genuine for granted, photographs are no longer considered as a "piece of truth". With the advance of digital image processing and computer graphics techniques, it has been easier than ever to modify images and forge new realities within minutes. Unfortunately, most of the times, these modifications seek to deceive viewers, change opinions or even affect how people perceive reality. In this context, this Ph.D. thesis (*`http://ic.unicamp.br/~tjose/publications/phd-thesis.pdf`*) builds upon the hypothesis that "image illumination inconsistencies are strong and powerful evidence of image composition" and presents four original and effective approaches to detecting image composition. They explore illumination inconsistencies in different ways to detect image composition forgery and together they bring to the forensic community important contributions which certainly will be a strong tool against image forgeries* [1].

1. Introduction

In a world where technology is improved daily at a remarkable speed, it is easy to face situations previously seen just on science fiction. One example is the use of advanced computational methods to solve crimes, an ordinary situation which usually occurs in TV shows such as the famous American crime drama television series *Crime Scene Investigation (CSI)*. However, technology improvements are, at the same time, a boon and a bane. Although it empowers people to improve their quality of life, it also brings huge drawbacks such as increasing the number of crimes involving digital documents, specially images. Image manipulations are present in almost all communication channels including newspapers, magazines, outdoors, TV shows, Internet, and even scientific papers [Rocha et al. 2011].

In this context, this work focuses on detecting one of the most common types of image manipulations: *splicing* or *composition*. Image splicing consists of using parts of two or more images to compose a new image that never took place in space and time. This composition process includes all the necessary operations (such as brightness and contrast adjustment, affine transformations, color changes, light matching, etc.) to construct realistic images able to deceive viewer.

[1]This Ph.D. thesis resulted from research performed by the student Tiago Carvalho advised by Dr. Anderson Rocha and Co-Advised by Dr. Hélio Pedrini.

After studying and analyzing the advantages and drawbacks of different types of methods for detecting image composition, our work herein relies on the following research hypothesis: **image illumination inconsistencies are strong and powerful evidence of image composition.**

This hypothesis has already been used by some researchers in the literature, and it is specially useful for detecting image composition because, even for expert counterfeiters, a perfect illumination matching is extremely hard to achieve. Also, there are some experiments that show how difficult it is for humans to perceive image illumination inconsistencies [Ostrovsky et al. 2005]. Due to these difficulties, it is impossible to trust just on expertise knowledge to detect image forgeries.

Therefore, the main objective of this work is to explore illumination inconsistencies to design and deploy efficient and effective methods for detecting image compositions, resulting as main scientific contribution the development of four new forensic techniques for detecting image composition.

The rest of this work is divided as follows: Section 2 presents our first actual contribution [Saboia et al. 2011] for detecting image composition, which is based on eye specular highlights. Section 3 describes our second contribution [Carvalho et al. 2013], result of a fruitful collaboration with researchers of the University of Erlangen-Nuremberg in Germany. The work is based on illuminant color characterization. Section 4 presents our third contribution [Carvalho et al. 2014a], result of a collaboration with other researchers at Unicamp and it is an improvement upon our work proposed in Section 3. Section 5 presents our last contribution [Carvalho et al. 2014b], result of a collaboration with researchers of Dartmouth College, US. This work uses the knowledge of users to estimate full 3-D light source position in images in order to point out possible forgery artifacts. Finally, Section 6 concludes the work putting our research in perspective and discussing new research opportunities.

2. Eye Specular Highlight Telltales for Digital Forensics

In this section, we introduce a new method for pinpointing image telltales in eye specular highlights to detect forgeries. Parts of the contents and findings in this section is published in the literature [Saboia et al. 2011].

Given a suspicious image, Johnson and Farid [Johnson and Farid 2008] proposed to apply a pre-processing in the image where the eye limbus regions are manually annotated and then estimates for each eye in the image the direction of light source, viewer (e. g., camera) and surface normal based on specular highlights present into the eye. Then, using a non-linear least squares function, such as Levenberg-Marquardt iteration, the method estimates a unique light source for the scene. Forgeries are detected analyzing the mean of angular error between scene light source and each eye light source. The proposed approach follows the pipeline illustrated in Figure 1.

In addition to the average angular error for light source position, we extend upon the previous work to take the estimation variation of this feature into account. Also, we estimate a unique viewer for the scene, calculating the mean and standard deviation of angular error between scene viewer and viewer estimated for each eye. These four features (mean of the angular errors relative to the light source (LME), variation of the

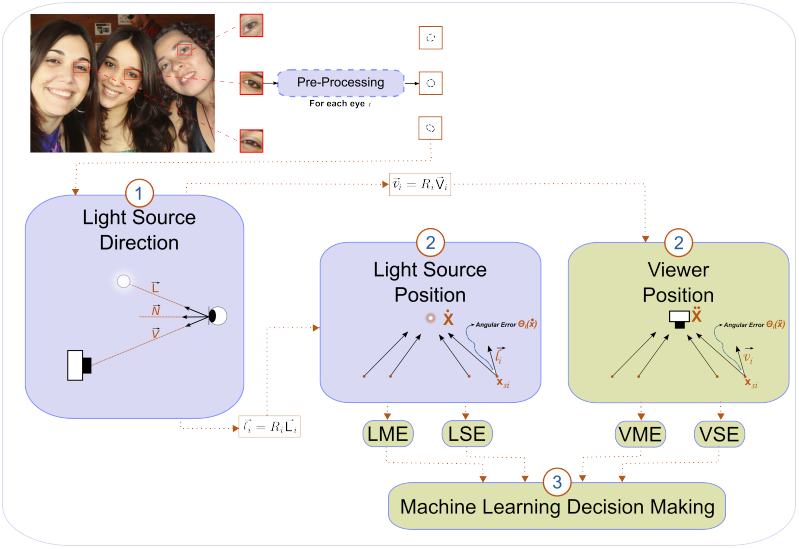


Figure 1. Proposed approach exploring eye specular highlights. Light green boxes indicate the extensions when compared with initial work.

angular errors relative to the light source (LSE), mean of the angular errors relative to the viewer (VME), variation of the angular errors relative to the light source (VSE)) compose a 4-d feature vector which is analyzed using a machine learning decision making approach (instead of a single hypothesis test as proposed in the initial work).

The newly-designed features and the new decision-making process reduce the classification error in more than 20% when compared to the prior work. To validate the ideas, we have used a data set of real composites and pristine (non-manipulated) images typically with more than three mega-pixels in resolution[2].

In summary, when compared against Johnson and Farid's [Johnson and Farid 2008] method, our method brings three main scientific contributions: (1) proposition of new discriminative features not explored previously; (2) use of machine learning approaches (single and multiple classifier combination and fusion) for the decision-making process instead of relying on a simple and limited hypothesis testing; and (3) reduction in the classification error in more than 20% when compared to the prior work.

It is worth noting, however, that the classification results are still affected by some drawbacks which inspired us to develop a new method described next.

3. Exposing Digital Image Forgeries by Illumination Color Classification

Different from our approach presented in Section 2, which is based on inconsistencies in the light setting, the approach proposed in this section relies on inconsistencies in light color.

Riess and Angelopoulou [Riess and Angelopoulou 2010] proposed to analyze illuminant color estimates from local image regions. Unfortunately, the interpretation of their resulting so-called *illuminant maps* is left to human experts. As it turns out, this decision is, in practice, often challenging. Thus, it is preferable to transfer the tampering

[2]http://ic.unicamp.br/∼rocha/pub/downloads/2014-tiago-carvalho-thesis/icip-eyes-database.zip

decision to an objective algorithm.

In this section, based upon Riess and Angelopoulou [Riess and Angelopoulou 2010] work, we propose important steps towards minimizing user interaction for an illuminant-based tampering decision-making. We propose a new semi-automatic method that is also significantly more reliable than earlier approaches. Parts of the contents and findings in this section is published in the literature [Carvalho et al. 2013].

We classify the illumination for each pair of faces in the image as either consistent or inconsistent. Throughout this section, we abbreviate illuminant estimation as IE, and illuminant maps as IM. The proposed method consists of five main components:

1. **Dense Local Illuminant Estimation (IE):** The input image is segmented into homogeneous regions. Per illuminant estimator, a new image is created where each region is colored with the extracted illuminant color. This resulting intermediate representation is called illuminant map (IM).
2. **Face Extraction:** This is the only step that may require human interaction. An operator sets a bounding box around each face (e. g., by clicking on two corners of the bounding box) in the questioned image. Alternatively, an automated face detector can be employed. We then crop every bounding box out of each illuminant map, so that only the illuminant estimates of the face regions remain.
3. **Computation of Illuminant Features:** for all face regions, texture-based and gradient-based features are computed on the IM values. Each one of them encodes complementary information for classification.
4. **Paired Face Features:** Our goal is to assess whether a pair of faces in an image is consistently illuminated. For an image with n_f faces, we construct $\binom{n_f}{2}$ joint feature vectors, consisting of all possible pairs of faces.
5. **Classification:** We use a machine learning approach to automatically classifying the feature vectors. We consider an image as a forgery if at least one pair of faces in the image is classified as inconsistently illuminated.

Quantitative evaluation shows that the proposed method achieves a detection rate of 79%, while existing automatic illumination-based work is slightly better than guessing. We exploit the fact that local illuminant estimates are most discriminative when comparing objects of the same (or similar) material. Thus, we focus on the automated comparison of human skin, and more specifically faces, to classify the illumination on a pair of faces as either consistent or inconsistent. User interaction is limited to marking bounding boxes around the faces in an image under investigation. In the simplest case, this reduces to specifying two corners (upper left and lower right) of a bounding box.

In summary, the main contributions of this work are: (1) interpretation of the illumination distribution as object texture for feature computation; (2) a novel edge-based characterization method for illuminant maps which explores edge attributes related to the illumination process; a large experiment with humans (not shown here) for evaluating their ability at pinpointing image forgeries; and (4) the creation of a benchmark dataset comprising 100 skillfully created forgeries and 100 original photographs[3].

[3]http://ic.unicamp.br/∼rocha/pub/downloads/2014-tiago-carvalho-thesis/tifs-database.zip

4. Splicing Detection via Illuminant Maps: More than Meets the Eye

In the previous section, we have introduced a new method based on illuminant color analysis for detecting forgeries on image compositions containing people. However, its effectiveness still needed to be improved for real forensic applications. Furthermore, some important telltales, such as illuminant colors, have not been statistically analyzed in the method. In this section, we introduce a new method for analyzing illuminant maps, which uses more discriminative features and a robust machine learning framework able to determine the most complementary set of features to be applied in illuminant map analyses. Parts of the contents and findings in this section are part of a submitted paper [Carvalho et al. 2014a][4].

We classify the illumination for each pair of faces in the image as either consistent or inconsistent. However, here we consider additional steps and more discriminative features. The following steps give an overview of the entire method:

1. **Dense Local Illuminant Estimation (IE):** similar to the the previous section, the first step of this method consists on segmenting the input image into homogeneous regions. Per illuminant estimator, a new image is created where each region is colored with the extracted illuminant color. This resulting intermediate representation is called illuminant map (IM).
2. **Choice of Color Space and Face Extraction:** before extracting people's faces from the IM, we first convert the IM to the desired color space (YC_bC_r, HSV, Lab and RGB). In contrast with Section 3, which has explored just YC_bC_r space, this conversion is useful given that some features are highlighted in certain color spaces. Afterwards, we extract people's faces.
3. **Feature Extraction from IM:** From each extracted face in the previous step, we now need to find telltales that allow us to correctly identify image splicing. Such information is present in different visual properties (e.g., texture, shape and color, among others) of the IM. In Section 3, proposed approach explores two properties, texture and edges, using one image descriptor per property. Here we also consider color information, which is an important property when visually analyzing IM maps. Finally, for each kind of information (e.g., texture, shape and color), we take advantage of several image description techniques and their complementarity to solve the problem.
4. **Face Characterization and Paired Face Features:** Given that in this section we consider more than one variant of IM, color space and description technique, $\mathcal{D}$ is an image descriptor composed of the triplet (IM, color space, and description technique). Assuming all possible combinations of such triplets according to the IM, color spaces and description techniques we consider herein, we have 54 different image descriptors. Thus, instead of analyzing each image face separately, after building $\mathcal{D}$ for each face in the image, we construct a feature vector $\mathcal{P}$ using the direct concatenation between two feature vectors $\mathcal{D}$ (provided by two faces). We classify this $\mathcal{P}$ vector.
5. **Face Pair Classification:** When using different IM, color spaces, and description techniques, the obvious question is how to automatically select the most important ones to keep and combine toward an improved classification performance.

[4]First round of revision.

For this purpose, we take advantage of a powerful classifier selection and fusion method Faria et al. [Faria et al. 2013].

6. **Forgery Detection:** Given an image I classified as fake, it is important to refine the analysis and point out which part of the image is actually the result of a composition. For that, we extract the IM using two different approaches and compare the color difference between them. This difference is higher, generally, in fake images than in pristine (non-manipulated) images. It allows us to train an SVM classifier which ultimately points out the face with highest probability to be the fake one.

The automatic forgery classification, in addition to the actual forgery localization, presented in this section represents an invaluable asset for forgery experts with a 94% classification rate, a remarkable 72% error reduction when compared to the method proposed in Section 3. Figure 2 depicts a direct comparison between the accuracy of both results as a bar graph.

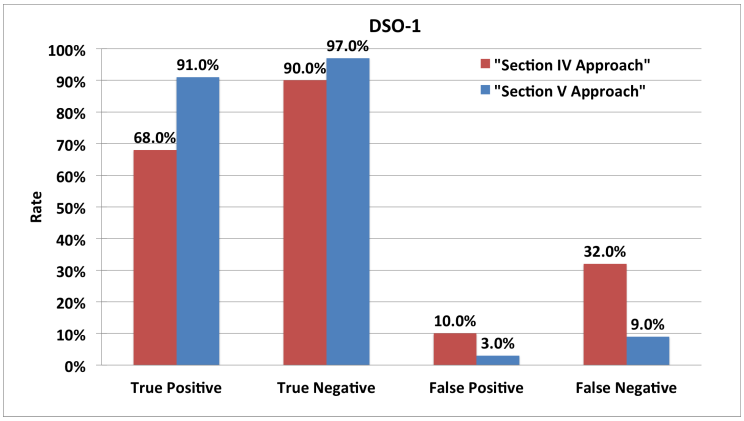


Figure 2. Comparison between results of methods presented in Section 3 and Section 4.

In summary, some of main contributions introduced in this method's section are: (1) the exploration of other color spaces not addressed in Section 3; (2) the incorporation of color descriptors, which showed to be very effective when characterizing illuminant maps; (3) a full study of the effectiveness and complementarity of many different image descriptors applied on illuminant maps to detect image illumination inconsistencies; (4) incorporation of a powerful machine learning framework to our approach, which automatically selects the best combination of all the factors of interest (e. g., IM, color space, descriptor, classifier); (5) the introduction of a new approach to detecting the most likely doctored part in fake images; (6) an evaluation on the impact of the number of IM and their importance to characterize an image in the composition detection task; and, finally, (7) an improvement of 15 percentage points in classification accuracy when compared to the results achieved with the method presented in Section 3, which represents a remarkable error classification reduction of 72%, as previously mentioned.

5. Exposing Photo Manipulation From User-Guided 3-D Lighting Analysis

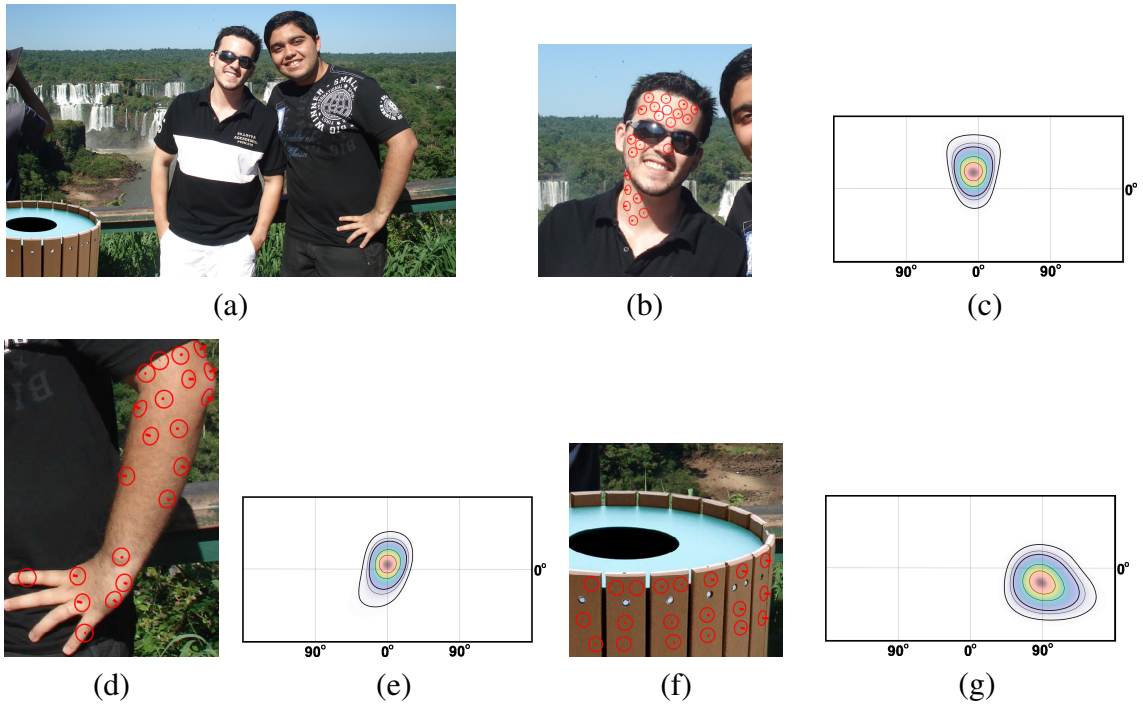
The approaches presented in the previous sections are specifically designed to detect forgeries involving people. However, sometimes an image composition can involve differ-

ent elements. A car or a building can be introduced into the scene with specific purposes. In this section, we introduce our last contribution, which focuses on detecting 3-D light source inconsistencies in scenes with arbitrary objects using a user's guided approach. Parts of the contents and findings in this section are part of a submitted paper [Carvalho et al. 2014b].

In the approach proposed in this section, we seek to estimate 3-D lighting from objects or people in a single image, relying on an analyst to specify the required 3-D shape from which lighting is estimated. To perform it, we describe a full work flow where first we use a user-interface for obtaining these shape estimates. This kind of approach has been chosen because there is good evidence from the human perception literature that human observers are fairly good at estimating 3-D shape from a variety of cues including, foreshortening, shading, and familiarity [Cole et al. 2009]. We have found that with minimal training, this task is relatively easy and accurate.

Secondly, we estimate 3-D lighting from these shape estimates, performing a perturbation analysis that contends with any errors or biases in the user-specified 3-D shape. Finally, we generate a region of possible light source position in terms of spherical coordinates. This region is very different (in terms of position) for pristine and fake objects at the same image.

Figure 3 depicts the result of approach application into a fake image.



(a) (b) (c)

(d) (e) (f) (g)

Figure 3. Different objects (b, d, f) and their respectively light source probability region (c, e, g) extracted from a fake image (a). The light source probability region (g) estimated for the fake object (f) is totally different from the light source probability region provided by the other objects.

In summary, this method presents two main contributions to the forensics community: (1) the possibility of estimating 3-D lighting properties of a scene from a single 2-D

image without knowledge of the 3-D structure of the scene; and (2) a study of user's skills on 3-D probes insertion for 3-D estimation of lighting properties in a forensic scenario.

6. Conclusion and Future Work

Along this work, we have presented four forensic techniques that work complementarily, developed to help forensic experts in their fight against image falsification. However, the main conclusion of this work is that forensic methods are in constant development. They have their pros and cons and there is no *"silver bullet"* able to detect all types of image composition with perfect classification results. The method described in Section 2 could not be applied to an image depicting a beach scenario, for instance, with people using sunglasses. However, this kind of image could be analyzed with the method proposed in Sections 3, 4 and 5. Similar analysis can be drawn for several other situations.

As research directions and future work, among all possible contributions, we would like to highlight a contribution for the approaches that explore illuminant colors. An interesting future work is the proposition of forms to compare illuminants provided by different body parts from the same person. This would remove the need of having two or more people in the image to detect forgeries and would be very useful for pornography image composition detection.

7. Related Publications

1. T. Carvalho, and A. Rocha, "Eye Specular Highlights Tell- tales for Digital Forensics: A Machine Learning Approach," in IEEE ICIP **(Conference - Qualis: A1 - Oral Presentation)**, 2011, pp. 1937–1940.
2. T. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, "Exposing digital image forgeries by illumination color classification," IEEE T.IFS **(Periodical - Impact Factor: 1.96)**, vol. 8, no. 7, pp. 1182–1194, 2013.
3. T. Carvalho, A. Pinto, E. Silva, F. da Costa, G. Pinheiro, and A. Rocha. ERI-MG – Anais da VII Escola Regional de Informática de Minas Gerais, chapter Crime Scene Investigation (CSI): da Ficção à Realidade **(Book Chapter - ISBN: 002316722X)**, pages 1 – 23. UFJF, 2012.
4. T. Carvalho, F. Faria, R. Torres, H. Pedrini, and A. Rocha.,"Splicing detection through illuminant maps: More than meets the eye" 2014, submitted to Elsevier Neurocomputing **(Periodical - Impact Factor: 1.81)**.
5. T. Carvalho, H. Farid, and E. Kee, "Exposing Photo Manipulation From User-Guided 3-D Lighting Analysis," 2015, submitted to IEEE SPIE//IS&T Electronic Imaging **(Conference with a big focus on forensics innovation)**.

References

[Carvalho et al. 2014a] Carvalho, T., Faria, F., Torres, R., Pedrini, H., and Rocha., A. (2014a). Splicing detection through color constancy maps: More than meets the eye. Submitted to Elsevier Forensics Science International (FSI).

[Carvalho et al. 2014b] Carvalho, T., Farid, H., and Kee, E. (2014b). Exposing Photo Manipulation From User-Guided 3-D Lighting Analysis. Submitted to IEEE International Conference on Image Processing (ICIP).

[Carvalho et al. 2013] Carvalho, T., Riess, C., Angelopoulou, E., Pedrini, H., and Rocha, A. (2013). Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security (T.IFS)*, 8(7):1182–1194.

[Cole et al. 2009] Cole, F., Sanik, K., DeCarlo, D., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S., and Singh, M. (2009). How well do line drawings depict shape? *ACM Transactions on Graphics (ToG)*, 28(3).

[Faria et al. 2013] Faria, F. A., dos Santos, J. A., Rocha, A., and da S. Torres, R. (2013). A framework for selection and fusion of pattern classifiers in multimedia recognition. *Pattern Recognition Letters*, 0(0):–.

[Johnson and Farid 2008] Johnson, M. K. and Farid, H. (2008). Exposing digital forgeries through specular highlights on the eye. In Furon, T., Cayre, F., Doërr, G. J., and Bas, P., editors, *ACM Information Hiding Workshop (IHW)*, volume 4567 of *Lecture Notes in Computer Science*, pages 311–325.

[Ostrovsky et al. 2005] Ostrovsky, Y., Cavanagh, P., and Sinha, P. (2005). Perceiving illumination inconsistencies in scenes. *Perception*, 34(11):1301–1314.

[Riess and Angelopoulou 2010] Riess, C. and Angelopoulou, E. (2010). Scene Illumination as an Indicator of Image Manipulation. In *ACM Information Hiding Workshop (IHW)*, volume 6387, pages 66–80.

[Rocha et al. 2011] Rocha, A., Scheirer, W., Boult, T. E., and Goldenstein, S. (2011). Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics. *ACM Computing Surveys*, 43(4):1–42.

[Saboia et al. 2011] Saboia, P., Carvalho, T., and Rocha, A. (2011). Eye Specular Highlights Telltales for Digital Forensics: A Machine Learning Approach. In *IEEE International Conference on Image Processing (ICIP)*, pages 1937–1940.

Two Approaches for Achieving Efficient Code-Based Cryptosystems

Rafael Misoczki[1]

[1]Project SECRET
INRIA - Institut National de Recherche en Informatique et Automatique
Université Pierre et Marie Curie

Supervisor: Nicolas Sendrier
PhD thesis defended on the 25th November, 2013 in Paris, France. Available at:
http://tel.archives-ouvertes.fr/docs/00/93/18/11/PDF/these_archivage_3073292.pdf

rmisoczki@larc.usp.br

Abstract. *Code-based cryptography (CBC) is one of the most prominent post-quantum alternatives to conventional cryptography. Although quantum-resistant and several times faster than its conventional counterparts, CBC is not widely deployed in practice. This is mostly due to its huge public-key sizes of several kilobytes. In this thesis, two different approaches to overcome this problem are introduced. One based on algebraic codes and another on graph-based codes. Effectively, in both cases, the public-key size is reduced to only a few kilobits, thus suppressing the main hindrance for the use of CBC in real-world applications. Moreover, under a quite reasonable assumption, we show that the security of the graph-based approach relies on a single well-studied problem. This is an important advantage in comparison with all CBC schemes and variants, including the classical McEliece scheme based on binary Goppa codes.*

1. Introduction and Motivation

Nowadays, cryptography is undoubtedly present everywhere. The necessity of secrecy, and more often privacy, is a crucial requirement for the modern world. Financial transactions, e-commerce and military applications are only a few examples that demonstrate the huge impact this research field has on our lives.

Due to its importance, many researchers have dedicated enormous amount of effort and time to propose and analyze efficient and secure cryptographic schemes. Number-theory cryptosystems, such as RSA [Rivest et al. 1978] and elliptic curves cryptosystems [Miller 1986], are good candidates and widely deployed in practice. Although providing a good trade-off between efficiency and security, they are not optimal in some other features. For example, they are vulnerable to attacks mounted with the help of quantum computers [Shor 1997].

This is not the case of cryptography based on coding-theory, where hard problems related to linear codes are exploited. Its security relies on the hardness of correcting errors in a seemingly random code and on distinguishing a public code from random. No quantum (nor classical) polynomial algorithm able to solve the decoding problem is known. On the other hand, the **code distinguishing problem** is usually the weakest pillar of CBC security, strongly depending on the choice of the code-family. In summary, it has been hard to associate the latter problem to the hardness of any well-studied problem.

Besides its post-quantum resistance, code-based cryptography has many virtues. It is several times faster than its number-theory counterparts and can be implemented using only very low complexity arithmetic. Nonetheless, it is not widely deployed in practice, mostly due to its important drawback: **huge public-key sizes**. To give an example, the classical McEliece scheme [McEliece 1978] is instantiated with binary Goppa codes and requires a public-key of 57 KB for 80 bits of security [Bernstein et al. 2008]. The security of this setup remains almost unscathed since its proposition. In the meantime, some attempts to reduce its key-sizes have been proposed. In most of the cases, those techniques have introduced fatal weaknesses to the scheme.

1.1. Objective of the Thesis

The objective of this thesis is twofold:

1. Reduce the code-based cryptosystems public-keys to sizes comparable to conventional cryptosystems, i.e. a few kilobits, without compromise its security.
2. Reduce the code distinguishing problem to some well-studied hard problem, thus strengthening its underlying security.

1.2. Organization of the Document

Section 2 introduces some basic concepts related to our work. Sections 3 and 4 describe our two approaches to achieve the aforementioned objectives. Each subsection ends with the indication where the result has been published. Section 5 presents our conclusions.

2. Preliminaries and Notation

In this section, the notation and some introductory concepts are presented.

2.1. Coding-Theory

Coding-theory gathers techniques to efficiently transmit information through channels subject to noise. This is done by correcting the noise at the receiver end. Equivalently, it means decoding the received message into the original one.

This goal is achieved by the employment of *codes*, which allow: encoding, decoding and, under certain conditions, error correction capability. Encoding is the process which maps a message, here called a *plaintext*, to an element of the code, here called a *codeword*. Decoding is a priori the process which performs the opposite, mapping a codeword to a plaintext. However, adding some redundancy during encoding, it might be possible to recover the plaintext even when a noisy codeword, called *word*, is received.

There are different types of codes. In this text we will be interested by linear codes. An (n, k)-linear code $\mathcal{C}$ is a vector subspace of length n and dimension k. Thus, codes can be defined and represented by matrices. A matrix $G \in \mathbb{F}_q^{k\times n}$ is called a *generator matrix* for a (n, k)-linear code $\mathcal{C}$ iff $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\}$. A matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ is called a *parity-check matrix* for a (n, k)-linear code $\mathcal{C}$ iff $\mathcal{C} = \{c \in \mathbb{F}_q^n \mid Hc^T = 0\}$. The Hamming weight, or simply weight, of a vector is the number of its non-zero coordinates.

2.2. Code-Based Cryptography

In this section, we describe the McEliece encryption scheme and we present a discussion regarding its security. The Niederreiter scheme has equivalent security [Li et al. 1994] and it is not hard to see that the same holds for the CFS digital signature [Courtois et al. 2001]. For this reason and without losing generality, we omit the details of these last two schemes and we will focus our attention on the McEliece scheme.

KEYGEN:

1. Generate a linear code $\mathcal{C} - (n, k)$ able to correct t errors.
   - Secret-Key: $\Psi_{\mathcal{C}}$, a t-error correcting procedure for $\mathcal{C}$.
   - Public-Key: $G \in \mathbb{F}_q^{k \times n}$, a systematic generator-matrix for $\mathcal{C}$.

ENC: The encryption of $m \in \mathbb{F}_q^k$ is:

1. Select at random a vector $e \in \mathbb{F}_q^n$ of weight t.
2. $x \leftarrow mG + e$.

DEC: The decryption of $x \in \mathbb{F}_q^n$ is:

1. Correct the errors of x: $x' \leftarrow \Psi_{\mathcal{C}}(x)$
2. $m \leftarrow$ Extract the first k positions of x'.

Table 1. McEliece Encryption Scheme.

2.2.1. Security of Code-Based Cryptography

In [Sendrier 2009], a security reduction for code-based public-key schemes is presented. A security reduction is a proof that an adversary able to attack the scheme is able to solve some (hard) computational problem with a similar effort. Let $\mathcal{S}_n(0, t)$ denote the sphere centered in zero of radius t in the Hamming space $\mathbb{F}_2^n$. Moreover, let $\mathcal{F}_{n,k}$ be a code family of (n, k)-linear codes able to correct t errors, $\mathcal{H}_{n,k}$ the set of all full rank matrices in $\mathbb{F}_2^{k \times n}$ and $\mathcal{K}_{n,k} \subset \mathcal{H}_{n,k}$ the public-key space of $\mathcal{F}_{n,k}$. The CBC security reduction problems are:

Problem 1 (Syndrome Decoding)**.**	**Problem 2** (Code Distinguishing)**.**
Parameters: $\mathcal{H}_{n,k}$*, an integer* $t > 0$.	*Parameters:* $\mathcal{K}_{n,k}$*,* $\mathcal{H}_{n,k}$.
Instance: $H \in \mathcal{H}_{n,k}$ *and* $s \in \mathbb{F}_2^r$.	*Instance:* $H \in \mathcal{H}_{n,k}$.
Problem: find $e \in \mathcal{S}_n(0, t)$ *s.t.* $eH^T = s$.	*Question: is* $H \in \mathcal{K}_{n,k}$*?*

Table 2. Underlying Security Problems of CBC.

Thus the two assumptions required by code-based cryptosystems to be *secure* are: the $(\mathcal{K}_{n,k}, \mathcal{H}_{n,k})$-code distinguishing problem is a *hard* problem and the $(\mathcal{H}_{n,k}, t)$-computational syndrome decoding problem is a *hard* problem.

3. Algebraic Approach: Quasi-p-adic Goppa Codes

Algebraic codes perform efficient decoding taking advantage from some algebraic structure, in general, consisting of polynomials over a ring. By evaluating such polynomials, it is possible to determine the error positions and error values present in a noisy codeword.

Goppa codes, an example of algebraic codes, are the most used codes in cryptography, mostly due to its resistance against structural attacks. Many attempts to reduce

the key-size of McEliece-like schemes relied on replacing Goppa codes by some other algebraic codes which admit compact representation. The vast majority of them has been successfully cryptoanalyzed. For this reason, our first approach is to find a way to reduce the key-sizes without leaving the Goppa code family.

Our first important remark come up with an alternative description of Goppa codes, which employs Cauchy matrices [Tzeng and Zimmermann 1975]. Let $q = p^m$ be a prime power. Cauchy matrices are defined by two disjoint sequences $z = (z_o, \ldots, z_{t-1}) \in \mathbb{F}_q^t$ and $L = (L_0, \ldots, L_{n-1}) \in \mathbb{F}_q^n$ of distinct elements, such that $C(z, L)_{ij} = 1/(z_i - L_j)$.

Definition 1 (Goppa code in Cauchy form)**.** *The Goppa code generated by a square-free polynomial $g(x) = (x - z_0) \ldots (x - z_{t-1})$ admits a parity-check matrix $H = C(z, L)$.*

Goppa codes in Cauchy form obviously provide a compact representation, since they are defined by two sequences instead of a whole matrix. However it is easy to note that such a code might be straightforwardly attacked. For this reason, we need to disguise it using some other matrix structure. Below we discuss our choice, the dyadic structure.

Definition 2 (Dyadic Matrix)**.** *Given a commutative ring $\mathcal{R}$ and a vector $h = (h_0, \ldots, h_{n-1}) \in \mathcal{R}^n$, the dyadic matrix $\Delta(h) \in \mathcal{R}^{n \times n}$ is the symmetric matrix with components given by $\Delta(h)_{ij} = h_{i \oplus j}$, where $\oplus$ denotes the bitwise XOR of the indices.*

In general, Cauchy matrices are not dyadic and vice-versa. However, as we will see, the intersection of these two classes of matrices is not empty, allowing the construction of Dyadic Goppa codes. Below we present our first contribution, a theorem that characterizes all Cauchy matrices in dyadic form.

Theorem 1 ([Misoczki and Barreto 2009])**.** *Let $H \in \mathbb{F}_q^{n \times n}$ with $n > 1$ be simultaneously a dyadic matrix $H = \Delta(h)$ for some $h \in \mathbb{F}_q^n$ and a Cauchy matrix $H = C(z, L)$ for two disjoint distinct sequences $z \in \mathbb{F}_q^n$ and $L \in \mathbb{F}_q^n$. Then $\mathbb{F}_q$ has characteristic 2, h satisfies:*

$$\frac{1}{h_{i \oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}, \tag{1}$$

and $z_i = 1/h_i + \omega$, $L_j = 1/h_j + 1/h_0 + \omega$ for some $\omega \in \mathbb{F}_q$.

Thus all we need is a method to solve Equation 1. Our second contribution is a linear-time algorithm to do so. Our technique consists of choosing distinct nonzero h_0 and h_i at random where i scans all powers of two smaller than n, and setting all other as

$$h_{i+j} \leftarrow \frac{1}{\frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}}$$

for $0 < j < i$ (so that $i + j = i \oplus j$), as long as this value is well-defined.

As before, a cryptosystem cannot be securely defined over a code specified by a parity-check matrix in Cauchy form. Thus we adopt similar techniques as presented in [Berger et al. 2009] for cyclic codes to hide its code-structure. In short, several operations are applied on the original Dyadic Goppa code that result in a block-shortened subfield subcode. The final code has a structure called quasi-dyadic, which is a (possibly non-dyadic) block matrix whose components are dyadic submatrices. This approach leads to very compact public-keys (9216 bits for 80-bits of security), without revealing its internal Goppa structure.

Contribution 1. The above result is published as: *Compact McEliece keys from Goppa codes*. R. Misoczki and P. S. L. M. Barreto. In Selected Areas in Cryptography (SAC2009), v. 5867 of Lecture Notes in Computer Science, pg. 376–392. Springer,2009.

3.1. Extending the Construction to the CFS Signature Scheme

The CFS scheme is only practical when instantiated with codes whose density of decodable syndromes is high. For example, consider a t-error correcting $\mathbb{F}_p$-alternant code, a code-family that encompasses Goppa codes, of length n derived from a code over $\mathbb{F}_{p^m}$. The syndrome space has size p^{mt}, but the decodable syndromes are only those that correspond to error vectors of weight not exceeding t. In other words, only $\sum_{w=1}^{t} \binom{n}{w}(p-1)^w$ nonzero syndromes are decodable and thus the density is $\delta = (1/p^{mt}) \sum_{w=1}^{t} \binom{n}{w}(p-1)^w$. For binary codes, this implies that the highest densities are attained only by nearly full length codes, i.e. $n \approx 2^m$. In this case, the density simplifies to $\delta \approx 1/t!$.

Unfortunately, our original construction for dyadic Goppa codes produces codes of fairly low density of decodable syndromes. The sequences z and L are distinct-disjoint, imposing an upper bounded on the code-length $n \leqslant 2^{m-1}$ and syndrome density $1/(2^t t!)$. For this reason, we have investigated techniques to relax the constraints imposed by our original construction in order to achieve better densities of decodable syndromes.

In this sense, we present a technique that permits some undefined entries in the dyadic matrix, since only a very few blocks are used in the final quasi-dyadic code. The result is an efficient algorithm that produces dyadic Goppa codes of much longer code-length ($2^m - t$, instead of 2^{m-1}) and affordable density $\delta \approx 0.5(t!)$.

Contribution 2. The above result is published as: *Quasi-dyadic CFS signatures*. P. S. L. M. Barreto, P.-L. Cayrel, R. Misoczki, and R. Niebuhr. In The 6th China International Conference on Information Security and Cryptology (Inscrypt 2010), volume 6584 of Lecture Notes in Computer Science, pages 336–349. Springer, 2010.

3.2. Generalizing the Construction for Codes of Characteristic $p \geq 2$

Most attempts at decreasing key sizes deal with codes in characteristic 2, in spite of evidence that odd characteristics may offer security advantages. In this section we explain how the approach started with quasi-dyadic codes can be generalized to codes defined over finite fields of characteristic greater than 2. We start by introducing p-adic matrices, which generalize dyadic matrices.

Definition 3 (p-adic Matrices)**.** *Let $A = \{a_0, \ldots, a_{N-1}\}$ be a finite abelian group of size $|A| = p^d$ with neutral element $a_0 = 0$, R a commutative ring and $h\colon A \longrightarrow R$ a sequence indexed by A. The p-adic matrix $M(h)$ is one for which $M_{i,j} = h(a_i - a_j)$ holds.*

Similarly to Theorem 1, we present a theorem that characterizes all p-adic matrices in Cauchy form. Moreover, we also present an efficient algorithm to construct p-adic Goppa codes. Those codes are efficiently decoded by the decoder presented in [Barreto et al. 2013], which is a generalization of Patterson's decoding algorithm. This construction leads to even more compact key-sizes, due to the fact that shorter code-lengths can be used for the same security level of dyadic Goppa codes.

Contribution 3. The above result is published as: *Monoidic codes in cryptography*. P. S. L. M. Barreto, R. Lindner, and R. Misoczki. In Post-Quantum Cryptography PQCrypto2011, v.7071 of Lecture Notes in Computer Science, p.179–199.Springer,2011.

3.3. Security Assessment

The p-adic Goppa proposal provides a comfortably large key space, invalidating any brute-force attack. On the other hand, structural attacks might have success if parameters are not conservatively chosen. This kind of attack attempts to recover the private code-structure from the public one.

The most dangerous structural attack has been presented in [Faugère et al. 2010]. The authors use the public generator matrix $G \in \mathbb{F}_q^{k\times n}$ and the fact that Goppa codes are also alternant codes to build a system of equations from the relation $GH^T = 0$ for an unknown alternant parity-check matrix H of the form: $H_{ij} = Y_i X_j^i$. This leads to a system of rn non-linear equations and $2n$ unknowns: $\{g_{i,0}Y_0X_0^j + \cdots + g_{i,n-1}Y_{n-1}X_{n-1}^j = 0 | i \in \{0,\ldots,k-1\}, j \in \{0,\ldots,r-1\}\}$.

This system cannot be solved for usual parameters of the original McEliece scheme using binary Goppa codes. However, the redundancy added by the dyadic (or cyclic) structure strongly decreases the number of unknowns of this system. This technique is effective against p-adic Goppa codes defined over intermediate fields. However, the attack fails when the codes are defined over the base field, as suggested in the thesis. In this case, the computation of the required Gröner basis is easy but trivial. It results in only one equation, not providing enough information to proceed with the attack. The algorithmic complexity of this attack is exponential, but a precise estimation of its cost remains an open problem.

4. Graph-Based Approach: Moderate-Density Parity-Check Codes

Graph-based codes are defined by a sparse parity-check matrix. They can also be seen as a bipartite graph (*a.k.a.* Tanner graph) taking the sparse parity-check matrix as its adjacency matrix. The best known graph-based codes are the Low-Density Parity-Check (LDPC) codes [Gallager 1963]. Its low-weight parity-check equations are used by belief-propagation techniques for efficient decoding, resulting in good correction performance.

LDPC codes cannot be directly used in a CBC scheme, since codeword finding algorithms could easily recover its low-weight equations and thus produce a decoding trapdoor [Monico et al. 2000]. This problem leads to a research topic aiming at disguising the private low-weight equations in the public code. One idea is to have the private code being an LDPC code and the public code being defined by another parity-check matrix obtained as the product of the LDPC parity-check matrix by some controlled weight matrix. This technique artificially increases the equations weight of the public code. Many McEliece variants based on this idea have been presented. The majority of them (except [Baldi et al. 2008]) have been broken due to the constraints imposed over the private auxiliary matrices.

Our first contribution is that these auxiliary matrices can be completely discarded by noticing the following remark. Instead of artificially increasing the public equations weight, it is more advantageous in terms of security and efficiency to simply use an LDPC code of increased weight. These are the Moderate-Density Parity-Check (MDPC) codes.

We suggest to have both private and public code being an MDPC code. The difference is that the private-key is a sparse code-description and the public-key is a dense description.

The benefits of employing MDPC codes are many. MDPC codes are only defined by some low-weight codewords existent in their dual code. Therefore it is natural to assume that the only way to distinguish such codes is by finding (or attesting the existence of) these low-weight dual codewords. Note that decoding is polynomially equivalent to finding low-weight codewords. Thus, under the aforementioned reasonable assumption, MDPC codes relies its security on a single well-studied problem. This is a remarkable advantage of our variant not only in comparison with compact-keys McEliece proposals, but also regarding the classical McEliece scheme instantiated with binary Goppa codes.

MDPC codes are decoded by using belief propagation techniques. This fact leads to two issues: decoding has a probabilistic nature (i.e. the decoding process is susceptible to fail) and poor error correction performance when working with codes of higher density.

Regarding its probability of decoding failure, a non-desired feature for cryptographic applications, three different approaches to address this issue are proposed. For the poor error correction capability, MDPC codes of very long code-length are required to correct a number of errors that thwarts generic decoding attacks. This particularity leads to key-sizes larger than the classical McEliece ones. To circumvent this problem, the addition of a quasi-cyclic structure is considered, leading to the family of QC-MDPC codes. This provides extremely compact public-keys of 4801 bits for 80-bits of security.

Note that the state of the art clearly indicates that a quasi-cyclic structure, by itself, does not imply a significant improvement for adversaries. All previous attacks on compact-key McEliece variants are based on the combination of a quasi-cyclic/dyadic structure with some *algebraic code information*, a characteristic absent in MDPC codes.

Contribution 4. The above result is published as: *MDPC-McEliece: New McEliece variants from moderate density parity-check codes*. In IEEE International Symposium on Information Theory (ISIT 2013), pages 2069–2073, Istambul, Turkey, 2013.

5. Conclusion

Code-based cryptosystems have been neglected to a second-class of cryptographic solutions for more than thirty years. Although efficient and secure against quantum attacks, its huge keys never permitted to be considered for practical applications. In this thesis, we presented two different approaches that significantly overcome this problem. Table 5 presents a comparison of the McEliece key sizes. The column Goppa refers to the classical McEliece scheme with binary Goppa codes, QC-LDPC to the previous proposal in the literature [Baldi et al. 2008], and p-adic and QC-MDPC refer to our two approaches. From the classical version to the QC-MDPC approach, a $\sim 98\%$ reduction is achieved.

Security	Goppa	QC-LDPC	p-adic Goppa	QC-MDPC
80	460 647	12 096	9 216	4 801

Table 3. Public-Key Size (in bits) Comparison

Moreover, the QC-MDPC proposal has a very remarkable feature. Under a quite

reasonable assumption, its security relies on a single well-studied problem. This an important advantage in comparison with all code-based public-key schemes and variants.

References

Baldi, M., Bodrato, M., and Chiaraluce, F. (2008). A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In *6th international conference on Security and Cryptography for Networks (SCN 2008)*, pages 246–262. Springer-Verlag.

Barreto, P. S., Misoczki, R., and Lindner, R. (2013). Decoding square-free goppa codes over $\mathbb{F}_p$. *IEEE Transactions on Information Theory*, 59(10):6851–6858.

Berger, T. P., Cayrel, P.-L., Gaborit, P., and Otmani, A. (2009). Reducing key length of the McEliece cryptosystem. In *Progress in Cryptology (AFRICACRYPT 2009)*, volume 5580 of *Lecture Notes in Computer Science*, pages 77–97. Springer.

Bernstein, D. J., Lange, T., and Peters, C. (2008). Attacking and defending the McEliece cryptosystem. In *Post-Quantum Cryptography (PQCrypto 2008)*, Lecture Notes in Computer Science, pages 31–46, Berlin, Heidelberg. Springer-Verlag.

Courtois, N., Finiasz, M., and Sendrier, N. (2001). How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology (Asiacrypt 2001)*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174, Gold Coast, Australia. Springer.

Faugère, J.-C., Otmani, A., Perret, L., and Tillich, J.-P. (2010). Algebraic cryptanalysis of McEliece variants with compact keys. In *Advances in Cryptology (EUROCRYPT 2010)*, volume 6110 of *Lecture Notes in Computer Science*, pages 279–298. Springer.

Gallager, R. G. (1963). *Low-Density Parity-Check Codes*. PhD thesis, M.I.T.

Li, Y. X., D., R. H., and W., X. M. (1994). On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Trans. on Inf. Theory*, 40(1):271–273.

McEliece, R. J. (1978). A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116.

Miller, V. S. (1986). Use of elliptic curves in cryptography. In *Advances in cryptology (CRYPTO 85)*, pages 417–426, New York, USA. Springer-Verlag.

Misoczki, R. and Barreto, P. (2009). Compact mceliece keys from goppa codes. In Jacobson, M. J. J., Rijmen, V., and Safavi-Naini, R., editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 376–392. Springer.

Monico, C., Rosenthal, J., and Shokrollahi, A. (2000). Using ldpc codes in the McEliece cryptosystem. In *IEEE Intl. Symp. on Inf. Theory (ISIT'2000)*, page 215. IEEE.

Rivest, R. L., Shamir, A., and Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of ACM*, 21(2):120–126.

Sendrier, N. (2009). On the use of structured codes in code based cryptography. In *Coding Theory and Cryptography III*, Contactforum, pages 59–68. Academie van België.

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509.

Tzeng, K. K. and Zimmermann, K. (1975). On extending Goppa codes to cyclic codes. *IEEE Transactions on Information Theory*, 21:721–716.

Autenticação e comunicação segura em dispositivos móveis de poder computacional restrito

Rafael Will Macedo de Araujo[1], **Routo Terada**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
Rua do Matão, 1010 - Butantã – São Paulo – SP – Brazil

{rwill,rt}@ime.usp.br

Abstract. *Protocols for authentication and key establishment are fundamental parts in security implementations for electronic devices communication. In applications involving devices with limited computational power communicating with a server, the choice of efficient protocols that require a simpler infrastructure is essential. In this work we implement secure key agreement protocols in ID-based and Certificateless public key cryptography models on ARM processor platforms. We also compare running times, memory and network usage.*

Resumo. *Protocolos de autenticação e de estabelecimento de chaves são peças fundamentais em implementações de segurança para comunicação de dispositivos eletrônicos. Em aplicações que envolvam dispositivos com poder computacional restrito comunicando-se com um servidor, é fundamental a escolha de protocolos eficientes e que necessitem de uma infraestrutura mais simples. Neste trabalho implementamos protocolos de acordo de chave seguros nos modelos de criptografia de chave pública baseado em identidade (ID-based) e sem certificado (Certificateless) em plataformas com processadores ARM. Comparamos tempos de execução, utilização de memória e uso do canal de comunicação.*

1. Introdução

O gerenciamento de certificados digitais requer uma infraestrutura de suporte mais robusta, o que acarreta em maiores custos de implementação e manutenção. Ademais, há uma série de dificuldades nos processos de de recuperação, validação e revogação de certificados. De maneira geral, os sistemas convencionais utilizam o criptossistema RSA, baseado no problema de fatoração de números inteiros grandes. Neste tipo de sistema as chaves públicas são números escolhidos aleatoriamente, e não possuem qualquer vínculo com o seu dono.

Os modelos de criptografia de chave pública baseado em identidade e sem certificado são alternativas ao modelo convencional de criptografia assimétrica. Em ambos os modelos a autenticação da chave pública ocorre implicitamente, durante a execução dos protocolos, sem a necessidade de gerenciar e distribuir certificados digitais. Tais sistemas, aliados ao uso de curvas elípticas, acarretam em menor custo operacional e computacional.

Neste contexto, o uso de criptografia de curvas elípticas se torna muito atraente para ambientes que dispõem menor capacidade de processamento. Através do uso de

*Este trabalho foi financiado pelo CNPq

*Dissertação de mestrado disponível em: http://www.ime.usp.br/~rwill/dissertacao

protocolos de acordo de chave com autenticação, é possível garantir confidencialidade, integridade e autenticidade de usuários e de equipamentos, sem prévia distribuição de segredos compartilhados, e estabelecer chaves compartilhadas entre dois usuários remotos, para uso posterior com algoritmos de criptografia simétrica.

2. Motivação e Contribuição

Dado um cenário onde um dispositivo com poder baixo poder computacional precisa se comunicar com um servidor, implementamos, de maneira eficiente, protocolos de acordo de chave nos modelos de criptografia baseado em identidade (*ID-Based*) [Shamir 1984], e sem certificado (*Certificateless*) [Al-Riyami and Paterson 2003]. Mostramos que é viável utilizar protocolos relativamente complexos, com níveis elevados de segurança, e tempos de execução praticáveis. Nossos testes simularam situações do mundo real, através de comunicação em uma rede TCP/IP, e utilização de bibliotecas criptográficas e matemáticas otimizadas, construídas principalmente em linguagens C e Assembly.

3. Problemas computacionais

A segurança dos protocolos analisados neste trabalho recai sobre quatro problemas computacionais, derivados do problema do logaritmo discreto em curvas elípticas. São eles:

- **DDH** (*Decision Diffie-Hellman Problem*): dados: $P, aP, bP, cP \in \mathbb{G}$; decidir: $abP = cP$.
- **CDH** (*Computacional Diffie-Hellman Problem*): dados: $P, aP, bP \in \mathbb{G}$; encontrar: abP.
- **GBDH** (*Gap-Bilinear Diffie-Hellman Problem*): dados: $P, aP, bP \in \mathbb{G}$; encontrar: abP, com ajuda de um oráculo de decisão (que dados $aP, bP, cP \in \mathbb{G}$, decide se $abP = cP$).
- **BDH** (*Bilinear Diffie-Hellman Problem*): dados: $P, aP, bP, cP \in \mathbb{G}$; encontrar: $e(P, P)^{abc}$.

Do ponto de vista teórico, um protocolo que tenha sido demonstrado seguro sob a hipótese de dificuldade do problema BDH é, pelo menos, tão seguro quanto outro que tenha sido demonstrado seguro sob a hipótese de dificuldade do problema Gap-BDH, no mesmo modelo de segurança. Por esse motivo, é preferível utilizar protocolos que se apoiem na suposição de dificuldade do problema BDH, e o uso do Gap-BDH deve ser evitado sempre que possível [Goya 2011].

4. Protocolos Analisados

A escolha dos protocolos a serem analisados levou em consideração os seguintes aspectos: modelo de segurança empregado, relevância do protocolo (quantidade de citações aos artigos em que são apresentados), ano de publicação (protocolos apresentados recentemente) e ausência de publicações científicas que comprovem falha de segurança ou quebra do protocolo. Outra exigência crucial foi a garantia de autenticidade dos usuários (isto é, protocolos de acordo de chave com autenticação, ou AKA - *Authenticated Key Agreement*).

Os protocolos escolhidos no modelo baseado em identidade foram demonstrados seguros sobre os modelos CK e eCK, definidos em [Canetti and Krawczyk 2001]

e [LaMacchia et al. 2007] respectivamente. Já protocolos escolhidos no modelo sem certificado foram demonstrados seguros sobre os modelos LBG (definido em [Lippold et al. 2009]), SJ$^+$, Mal-LBG e Mal-SJ$^+$ (definidos em [Goya 2011]). Todos são extensões do modelo eCK que incluem adaptações para o caso sem certificado. Na sequência, são listados os protocolos de acordo de chave escolhidos, o problema computacional e o modelo de segurança no qual se baseiam. Maiores detalhes sobre a descrição dos protocolos e suas operações estão disponíveis em [Araujo 2013].

Protocolos no modelo baseado em identidade:

- **HC-BDH**: seguro no modelo eCK, [Huang and Cao 2009]
- **HLZ-GBDH**: seguro no modelo eCK, [Hu et al. 2009]
- **CC-BDH**: seguro no modelo CK, [Chow and Choo 2007]
- **NCLH-BDH** e **NCLH-GBDH**: seguro no modelo eCK, [Ni et al. 2011]
- **NCL-BDH**: seguro no modelo eCK, [Ni et al. 2012]

Protocolos no modelo sem certificado:

- **LBG-BDH** e **LBG-GBDH**: seguro no modelo LBG, [Lippold et al. 2009]
- **GOT-BDH** e **GOT-GBDH**: seguro no modelo LBG, [Goya et al. 2010]
- **GNT3-BDH**: seguro no modelo LBG, [Goya 2011]
- **GNT1-GBDH**: seguro no modelo Mal-LBG, [Goya et al. 2011]
- **GNT2-GBDH**: seguro no modelo Mal-SJ$^+$, [Goya 2011]
- **GNT4-BDH**: seguro no modelo SJ$^+$, [Goya 2011]

5. Experimentos

O cenário proposto para os experimentos consiste em acordar uma chave secreta entre um dispositivo de baixo poder computacional (cliente) e um servidor, que funciona como a autoridade do sistema, conforme ilustrado na Figura 1. Neste ambiente não há distribuição prévia de segredos compartilhados. A comunicação de ambos acontece por meio de uma rede TCP/IP, utilizando os padrões IEEE 802.11g (para comunicação sem fio) e 100BASE-TX (*Fast Ethernet*), a depender do dispositivo.

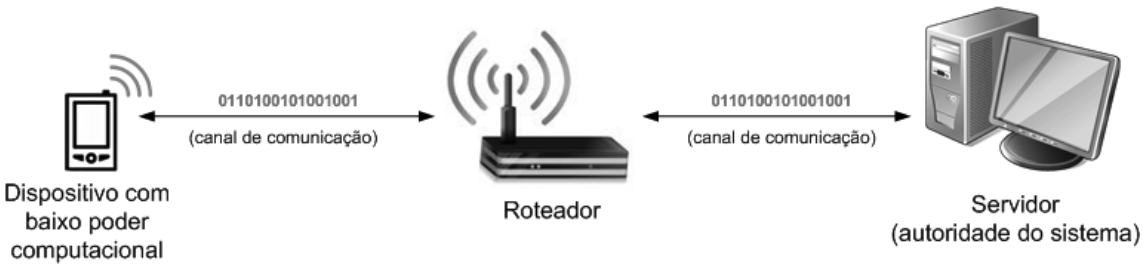


Figura 1. Cenário de comunicação entre o dispositivo e o servidor (autoridade do sistema) [Araujo and Terada 2013]

Os experimentos foram realizados utilizando-se três dispositivos como clientes: um *smartphone*, um *tablet*, e um *single-board computer* voltado para sistemas embarcados. Tais dispositivos dispõem de sistema operacional Android ou Linux. A seguir são detalhadas as configurações dos equipamentos utilizados:

- **Servidor (PC)**: processador Intel Core 2 Duo T5800 de 2.0 GHz, memória RAM de 3 GB, e sistema operacional Ubuntu 12.04 LTS de 32 bits.

- **Smartphone (MM1)**: Motorola Milestone 1, processador ARM Cortex-A8 (*single-core*) de 600 MHz, memória RAM de 256 MB, e sistema operacional Android 2.2.
- **Tablet (GN7)**: Asus Google Nexus 7, processador ARM Cortex-A9 (*quad-core*) de 1.2 GHz, memória RAM de 1 GB, e sistema operacional Android 4.2.
- **Single-board computer (RPi)**: Raspberry Pi, processador ARM1176JZF-S (*single-core*) de 700 MHz, memória RAM de 256 MB, e sistema operacional Debian "Wheezy" (armhf).

5.1. Ambiente de desenvolvimento

Escolhemos a *RELIC-toolkit* (versão 0.3.1) [Aranha and Gouvêa 2009] como biblioteca criptográfica, construída em linguagens C e Assembly. Consequentemente, as implementações dos protocolos estudados foram escritas em linguagem C, integrando-se de forma transparente à biblioteca *RELIC-toolkit*. A comunicação entre os dispositivos e o servidor é realizada através de *sockets* em C. Dentre as vantagens da utilização de *sockets* estão a fácil integração dos códigos construídos com as funções de comunicação em rede, comunicação TCP/IP direta, e facilidade em transmitir alguns tipos de dados complexos da *RELIC-toolkit*, já que não existem funções nativas que convertam tais tipos para *strings* e vice-versa.

5.2. Curvas elípticas e custo das operações

A biblioteca *RELIC-toolkit* dispõe de um conjunto de curvas elípticas pré-definidas. Tais curvas se dividem em dois grupos: curvas elípticas sobre corpos binários e curvas elípticas sobre corpos primos. Nos interessam as curvas conhecidas como *pairing-friendly*, que permitem a construção de sistemas criptográficos que fazem uso de emparelhamentos bilineares [Freeman et al. 2010].

Foram selecionadas seis curvas elípticas que oferecem diferentes níveis de segurança, sendo três definidas sobre corpos binários e três definidas sobre corpos primos. As curvas binárias escolhidas possuem grau de mergulho 4, e estão definidas sobre corpos binários de 271, 353 e 1223 bits. Estas curvas trabalham com emparelhamento simétrico. No caso das curvas primas, foram selecionadas curvas elípticas conhecidas como BN [Barreto and Naehrig 2006], definidas sobre corpos primos de 158, 256 e 638 bits. Estas curvas trabalham com emparelhamento assimétrico e possuem grau de mergulho 12, permitindo assim o uso de corpos algébricos menores, contudo, mantendo elevado nível de segurança. Por este motivo, são mais eficientes quando comparadas com curvas binárias.

Para mensurar o desempenho de cada uma dessas curvas, medimos o tempo de execução das principais operações envolvidas nos protocolos analisados, conforme a Tabela 1. Para curvas primas, faz-se necessário mensurar operações em $\mathbb{G}_1$ e em $\mathbb{G}_2$, já que $\mathbb{G}_1 \neq \mathbb{G}_2$. Foram utilizadas 50 amostras de tempo para o cálculo de cada um dos valores apresentados.

Tabela 1. Intervalo de confiança (95%) dos tempos de execução (em milissegundos) das operações para diferentes níveis de segurança

		Nível de segurança das curvas elípticas binárias		
Operação	**Dispositivo**	**70 bits**	**80 bits**	**128 bits**
Emparelhamento ηT	PC	[5.14, 5.15]	[8.61, 8.61]	[117.0, 117.15]
	RPi	[45.46, 45.57]	[79.46, 79.61]	[1229.19, 1229.56]
	MM1	[42.16, 42.56]	[72.9, 73.43]	[1275.74, 1295.76]
	GN7	[16.44, 16.48]	[51.87, 51.97]	[479.47, 479.98]
Exponenciação em $\mathbb{G}_T$	PC	[5.93, 6.06]	[10.63, 10.81]	[149.97, 151.74]
	RPi	[52.45, 53.56]	[98.86, 100.48]	[1572.75, 1591.09
	MM1	[50.06, 51.2]	[93.94, 95.85]	[1641.15, 1666.62]
	GN7	[19.45, 19.89]	[66.42, 68.12]	[613.84, 621.1]
Multiplicação em $\mathbb{G}_T$	PC	[0.04, 0.04]	[0.06, 0.06]	[0.23, 0.23]
	RPi	[0.38, 0.39]	[0.52, 0.53]	[2.42, 2.47]
	MM1	[0.35, 0.36]	[0.47, 0.51]	[2.45, 2.97]
	GN7	[0.14, 0.14]	[0.33, 0.34]	[0.96, 0.96]
Multiplicação em $\mathbb{G}$	PC	[3.44, 3.48]	[6.09, 6.15]	[75.05, 75.53]
	RPi	[30.81, 31.15]	[56.0, 56.57]	[781.48, 785.98]
	MM1	[29.38, 31.82]	[53.65, 54.42]	[828.44, 838.29]
	GN7	[11.5, 11.63]	[37.73, 38.26]	[298.53, 300.18]
Adição em $\mathbb{G}$	PC	[0.03, 0.03]	[0.03, 0.04]	[0.13, 0.13]
	RPi	[0.25, 0.26]	[0.32, 0.33]	[1.43, 1.45]
	MM1	[0.22, 0.23]	[0.29, 0.3]	[1.44, 1.9]
	GN7	[0.09, 0.09]	[0.21, 0.21]	[0.56, 0.56]

		Nível de segurança das curvas elípticas primas		
Operação	**Dispositivo**	**78 bits**	**128 bits**	**192 bits**
Emparelhamento *Optimal Ate*	PC	[5.47, 5.48]	[13.07, 13.08]	[90.06, 90.08]
	RPi	[42.22, 42.33]	[94.66, 94.83]	[624.75, 625.04]
	MM1	[49.34, 50.67]	[110.67, 112.57]	[684.97, 695.99]
	GN7	[31.37, 31.92]	[68.14, 68.19]	[408.1, 409.0]
Exponenciação em $\mathbb{G}_T$	PC	[5.27, 5.35]	[12.59, 12.74]	[88.07, 88.74]
	RPi	[40.88, 41.54]	[91.04, 92.17]	[612.32, 616.96]
	MM1	[48.15, 49.64]	[107.83, 110.25]	[673.39, 685.02]
	GN7	[30.49, 31.25]	[66.42, 67.27]	[401.86, 405.3]
Multiplicação em $\mathbb{G}_T$	PC	[0.04, 0.04]	[0.05, 0.05]	[0.15, 0.15]
	RPi	[0.27, 0.27]	[0.37, 0.38]	[1.02, 1.06]
	MM1	[0.27, 0.27]	[0.36, 0.64]	[0.98, 1.24]
	GN7	[0.18, 0.19]	[0.25, 0.26]	[0.66, 0.66]
Multiplicação em $\mathbb{G}_1$	PC	[0.56, 0.56]	[1.4, 1.4]	[10.01, 10.06]
	RPi	[4.5, 4.54]	[10.14, 10.2]	[70.32, 70.65]
	MM1	[4.91, 5.46]	[11.03, 11.75]	[77.9, 79.63]
	GN7	[3.5, 3.57]	[7.55, 7.6]	[45.6, 46.47]
Multiplicação em $\mathbb{G}_2$	PC	[2.35, 2.38]	[6.07, 6.13]	[47.31, 47.62]
	RPi	[16.98, 17.22]	[41.25, 41.7]	[316.97, 319.02]
	MM1	[19.78, 20.78]	[48.25, 51.73]	[348.66, 354.51]
	GN7	[12.77, 13.06]	[29.57, 29.89]	[206.17, 207.75]
Adição em $\mathbb{G}_1$	PC	[0.004, 0.004]	[0.01, 0.01]	[0.02, 0.02]
	RPi	[0.03, 0.03]	[0.04, 0.05]	[0.12, 0.12]
	MM1	[0.04, 0.05]	[0.05, 0.05]	[0.09, 0.32]
	GN7	[0.02, 0.02]	[0.03, 0.03]	[0.07, 0.08]
Adição em $\mathbb{G}_2$	PC	[0.01, 0.01]	[0.01, 0.01]	[0.04, 0.04]
	RPi	[0.07, 0.07]	[0.1, 0.1]	[0.26, 0.28]
	MM1	[0.07, 0.07]	[0.1, 0.11]	[0.23, 0.41]
	GN7	[0.05, 0.05]	[0.06, 0.06]	[0.17, 0.17]

Utilizamos a biblioteca GMP (GNU *Multiple Precision Library*) como *backend* aritmético da *RELIC-toolkit*. Desta forma, o tempo médio das operações diminuiu na razão de 3 a 8 vezes, dependendo do dispositivo e do nível de segurança. Pode-se observar, pela Tabela 1, que as operações que demandam mais tempo são emparelhamento, exponenciação em $\mathbb{G}_T$ (que em curvas binárias chega a ser mais custosa que emparelhamento), multiplicação escalar nos grupos $\mathbb{G}$ (curvas binárias) e em $\mathbb{G}_2$ e $\mathbb{G}_1$ (curvas primas).

Observa-se que o desempenho das curvas primas foi muito superior ao das curvas binárias, mesmo as primeiras apresentando um nível de segurança mais elevado. Devido à existência de dois grupos algébricos nos corpos de característica prima, podemos utilizar a propriedade de bilinearidade dos emparelhamentos para converter operações de exponenciação no grupo $\mathbb{G}_T$ em multiplicações nos grupos $\mathbb{G}_1$ ou $\mathbb{G}_2$, resultando em um ganho de eficiência maior que 50% para este tipo de conversão.

Outra observação relevante é que o problema do logaritmo discreto em corpos de característica pequena (que é o caso das curvas binárias utilizadas) foi criptoanalisado recentemente em [Barbulescu et al. 2014]. Desta maneira, além de menos eficientes que as curvas primas, o uso de curvas elípticas binárias é considerado inseguro atualmente.

5.3. Adaptação eficiente para emparelhamento assimétrico

Uma forma de implementar protocolos mais eficientes, ao utilizar curvas primas, é distribuir os valores dos grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ de forma que o dispositivo com menor poder computacional (cliente) realize mais cálculos no grupo $\mathbb{G}_1$, enquanto o servidor (que possui maior poder computacional) realize mais operações no grupo $\mathbb{G}_2$. Como visto na Tabela 1, operações em $\mathbb{G}_2$ chegam a custar em média quatro vezes mais que operações em $\mathbb{G}_1$, e por este motivo devemos maximizar a quantidade de multiplicações em $\mathbb{G}_1$ no lado do cliente.

Em [Araujo and Terada 2013] foram apresentadas duas adaptações do protocolo Huang-Cao [Huang and Cao 2009] para emparelhamento assimétrico. Na primeira delas, o servidor terá que executar duas multiplicações em $\mathbb{G}_1$ e duas multiplicações em $\mathbb{G}_2$, e o cliente executará três multiplicações em $\mathbb{G}_2$, que são mais custosas. Já o segundo cenário é mais vantajoso para o cliente, que calculará apenas três multiplicações em $\mathbb{G}_1$, e o servidor continua com duas multiplicações em $\mathbb{G}_1$ e duas multiplicações em $\mathbb{G}_2$. Adaptações similares a esta foram produzidas para todos os protocolos implementados, com o objetivo de tornar o código utilizado pelos dispositivos o mais eficiente possível.

5.4. Desempenho dos protocolos analisados

Em [Araujo 2013] e [Araujo and Terada 2013] são apresentadas comparações entre os tempos de execução dos protocolos estudados, nos três dispositivos analisados, para as seis curvas elípticas selecionadas. A contagem de tempo iniciou-se no momento da escolha dos valores temporários, e encerrou-se após o cálculo da chave de sessão. Para cada protocolo, em cada dispositivo, foram obtidas um total de 100 amostras de tempo, sendo 50 para o caso sem pré-computação, e 50 para o caso com pré-computação. Para cada conjunto de 50 amostras calculou-se a média e o intervalo de confiança (de 95%), sendo que cada amostra corresponde a uma sessão completa, ou seja, escolhem-se novos valores temporários e calcula-se uma nova chave de sessão, diferente das anteriores. O cenário sem pré-computação representa a execução dos protocolos da maneira como são definidos. Já o cenário com pré-computação considera que uma entidade A se comunica frequentemente com outra entidade B e, desta forma, alguns valores referentes a B não precisam ser recalculados a cada nova sessão.

Os resultados confirmam que protocolos sob modelos de segurança mais fortes e problemas computacionais mais difíceis são mais lentos. Alguns protocolos são versões aprimoradas de outros um pouco mais antigos, como é o caso do NCLH-BDH e NCL-BDH, e dos protocolos GNT3, GOT e LBG. Nestes casos houve melhoria de eficiência ou

equivalência, sendo que nos casos onde o tempo é equivalente o consumo de memória foi menor. Também em [Araujo 2013] e [Araujo and Terada 2013] são apresentadas tabelas que comparam a quantidade de memória RAM utilizada e o comprimento das mensagens trocadas (ambos em *bytes*) por protocolo e curva elíptica.

6. Conclusão

Este trabalho propôs implementações eficientes de protocolos de acordo de chave em dispositivos de poder computacional restrito nos modelos de criptografia baseado em identidade e sem certificado. Os resultados dos experimentos mostram que é possível utilizar tais protocolos com tempos praticáveis. Apontamos sugestões para desenvolvedores de como implementar protocolos de forma eficiente, através da redistribuição das operações de multiplicação nos grupos $\mathbb{G}_1$ e $\mathbb{G}_2$, quando utilizado emparelhamento assimétrico, ou substituição de operações de exponenciação no grupo $\mathbb{G}_T$ por multiplicações em $\mathbb{G}$ quando a troca é de um para um. Também verificou-se o uso de memória RAM e comprimento das mensagens trocadas que, juntamente com as comparações de tempo, formam um guia para desenvolvedores que pretendam utilizar algum desses protocolos em situações reais. As contribuições deste trabalho resultaram na publicação de [Araujo 2013].

Referências

Al-Riyami, S. S. and Paterson, K. G. (2003). Certificateless public key cryptography. In *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, Taipei, Taiwan. Springer.

Aranha, D. F. and Gouvêa, C. P. L. (2009). RELIC is an Efficient LIbrary for Cryptography. `http://code.google.com/p/relic-toolkit/`.

Araujo, R. W. M. (2013). Autenticação e comunicação segura em dispositivos móveis de poder computacional restrito. Master's thesis, Instituto de Matemática e Estatística da Universidade de São Paulo.

Araujo, R. W. M. and Terada, R. (2013). Implementação eficiente de protocolos de acordo de chave em dispositivos de poder computacional restrito. *SBSeg 2013, XIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 323–336.

Barbulescu, R., Gaudry, P., Joux, A., and Thomé, E. (2014). A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology–EUROCRYPT 2014*, pages 1–16. Springer.

Barreto, P. S. and Naehrig, M. (2006). Pairing-friendly elliptic curves of prime order. In *Selected areas in cryptography*, pages 319–331. Springer.

Canetti, R. and Krawczyk, H. (2001). Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology-EUROCRYPT 2001*, pages 453–474. Springer.

Chow, S. S. and Choo, K.-K. R. (2007). Strongly-secure identity-based key agreement and anonymous extension. In *Information Security*, pages 203–220. Springer.

Freeman, D., Scott, M., and Teske, E. (2010). A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280.

Goya, D., Nakamura, D., and Terada, R. (2011). Acordo de chave seguro contra autoridade mal intencionada. *SBSeg 2011, XI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 265–278.

Goya, D., Okida, C., and Terada, R. (2010). A two-party certificateless authenticated key agreement protocol. *SBSeg 2010, X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 443–446.

Goya, D. H. (2011). *Criptografia de chave pública sem certificado*. PhD thesis, Instituto de Matemática e Estatística da Universidade de São Paulo.

Hu, X., Liu, W., and Zhang, J. (2009). An efficient id-based authenticated key exchange protocol. In *Information Engineering, 2009. ICIE'09. WASE International Conference on*, volume 2, pages 229–233. IEEE.

Huang, H. and Cao, Z. (2009). An id-based authenticated key exchange protocol based on bilinear diffie-hellman problem. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 333–342. ACM.

LaMacchia, B., Lauter, K., and Mityagin, A. (2007). Stronger security of authenticated key exchange. In *Provable Security*, pages 1–16. Springer.

Lippold, G., Boyd, C., and Gonzalez Nieto, J. (2009). Strongly secure certificateless key agreement. In *Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography*, Pairing '09, pages 206–230, Berlin, Heidelberg. Springer-Verlag.

Ni, L., Chen, G., and Li, J. (2012). Escrowable identity-based authenticated key agreement protocol with strong security. *Computers & Mathematics with Applications*.

Ni, L., Chen, G., Li, J., and Hao, Y. (2011). Strongly secure identity-based authenticated key agreement protocols. *Computers & Electrical Engineering*, 37(2):205–217.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, volume LNCS 196, pages 47–53. Springer-Verlag New York, Inc.

Malware Behavior

André Grégio[1], Mario Jino (Orientador)[2], Paulo Lício de Geus (Co-Orientador)[3]

[1]Divisão de Segurança de Sistemas de Informação (DSSI)
Centro de Tecnologia da Informação Renato Archer (CTI) – Campinas – SP – Brazil

[2]Faculdade de Engenharia Elétrica e da Computacão (FEEC)
Universidade Estadual de Campinas (Unicamp) – Campinas – SP – Brazil

[3]Instituto de Computação (IC)
Universidade Estadual de Campinas (Unicamp) – Campinas– SP – Brazil

{gregio,paulo}@lasca.ic.unicamp.br, jino@dca.fee.unicamp.br

Abstract. *Malware attacks are the most dangerous current threat to computer systems security. The main mechanism used for protection against malware is the antivirus, which does not provide sufficient information about the infection and may be easily bypassed by obfuscation and anti-analysis techniques. Therefore, we need to deeply understand what malware samples do during an attack so as to develop effective defense mechanisms. In this work, we delve into malware behavior to propose (i) an extensible, behavior-centric taxonomy, (ii) a dynamic analysis system that extracts behavioral profiles, (iii) detection techniques for Internet Banking malware, (iv) a visualization tool for execution traces, and (v) an instruction-based clustering technique to identify families and code reuse.*

Resumo. *O ataque por* malware *é uma das ameaças mais perigosas para a segurança de sistemas. O principal mecanismo de proteção utilizado é o antivírus, que não provê informações suficientes sobre a infecção e pode ser facilmente superado por técnicas de ofuscação e anti-análise. Logo, é necessário um entendimento profundo sobre as atividades de* malware *durante ataques para que se desenvolva mecanismos de defesa efetivos. Neste trabalho, explora-se o comportamento de* malware *e propõe-se (i) uma taxonomia estensível baseada em comportamento, (ii) um sistema de análise dinâmica que extrai perfis comportamentais, (iii) técnicas para detecção de* malware *para Internet Banking, (iv) ferramentas de visualização de traços de execução e (v) uma técnica de agrupamento baseada em instruções para identificar famílias e reuso de código.*

1. Introduction

The spread of malicious programs through computer networks, mainly the Internet, has been a major threat to the security of interconnected systems. Malicious programs, commonly referred to as **malware**, can be understood as applications whose intent is to compromise a system. Those applications are commonly named as viruses, worms, Trojans, backdoors, keyloggers, and so on. One of the greatest motivations for malware attacks is the underground economy that is already established [Fallmann et al. 2010] [Holz et al. 2009] [Stone-Gross et al. 2011b], based on compromised infrastructures renting (e.g., network-connected systems that are invaded

and remote controlled by attackers, such as botnets), sensitive information stealing (e.g., Internet Banking credentials, usernames and passwords of e-mail accounts, credit card numbers) [Stringhini et al. 2012], unsolicited messages (e.g., spam, fake product offers) [Levchenko et al. 2011] [Stone-Gross et al. 2011a] and advertisement clicking [Lauinger et al. 2012]. Thus, the identification of a program as being a known malware (already collected and analyzed, maybe defeated) allows for efficient and effective incident response. The countermeasures taken, for its part, can facilitate the damage containment process, decrease losses, and mitigate side infections through security blocking rules and patch application.

Currently, one of the most popular defense mechanism against malware is still the antivirus (AV). However, the major issue regarding AV engines is the frequent and increasing rise of malware variants. Malware variants correspond to previously identified malware families modified until they either do not match a known signature or are able to evade, compromise or subvert the protection mechanisms to remain stealthy. Another issue that must be taken into account is that malware developers have been embedding self-defense mechanisms to their products, i.e. current malware may disable the operating system native protection (e.g., firewall, AV, security plugins, updates), verify if it is under some kind of analysis and do not present its malicious behavior (e.g., by modifying its execution on-the-fly), be packed in a way that avoids analysis and detection (e.g., checking its integrity in memory), disguise itself as a system application, a legitimate software, or a fake antivirus, and so on.

Apart from those aforementioned issues, the AV developers' community is still not tied to a common standard to classify detected malware samples. This slows malware-related incident response procedures, turning them ineffective in certain cases. Nowadays, the boundaries that divide a malware class from another do not exist anymore, since modern malware samples are usually built on functional modules that exhibit, at the same time, the behavior expected from rootkits, Trojan horses, worms, viruses, flooders, spammers and botclients. In this thesis, we address issues related to malware behavior and provide the contributions mentioned in Section 2.

1.1. Objectives

The main goal of this thesis is to provide means to identify malicious behavior in unknown programs, i.e., those not detected by antivirus. To do so, we studied how malware behaves (using a practical approach) and how to use the information acquired from the observed behaviors. The aim is to develop detection techniques, as well as a more meaningful classification scheme, in order to handle the damage malware can cause to an infected system. To this end, forwarding this point, we present the research work done in the context of a Ph.D. thesis as a proposal of addressing the aforementioned needs, aiming to respond to malware incidents in a useful, organized and understandable manner. This is accomplished through the extraction of malicious behavior using a dynamic analysis system that we have been developing, the proposal of a behavior-based taxonomy for classification, the addition of modules for the detection of specialized malware (bankers) and visualization of malware execution traces, and, finally, the introduction of a clustering algorithm to address malware behavior at instruction level.

2. Contributions

The field of malware research is very broad and plenty of effort has been spent by the security community to address the several types of threats that malicious code poses to Internet-connected systems. In this thesis, the focus is on malware behavior and in what can be done with them concerning computer systems defense. Thus, in addition to defining dangerous behaviors and malware classes based on them, we also apply the behavioral profiling on other topics, such as detection, classification, clustering, code reuse identification, visualization and incident response. The main contributions of this thesis are:

- A brief review of the history of malicious programs, a discussion about the current malware naming scheme issues and antivirus labeling, and a survey on the diversity of malware taxonomies according to their types (e.g., worms, bots).
- A definition of the different types of behavior that a malicious program can present, the description of a set of dangerous activities gathered from actually analyzed samples, and the proposal of a behavior-centric malware taxonomy.
- A malware dynamic analysis system that inspects for suspicious behaviors and extracts behavioral profiles from monitored programs.
- An approach to detect information stealing malware that leverages a subset of the defined behaviors and image processing techniques.
- Interactive visualization tools to help in identifying similar behavioral patterns.
- A heuristic to cluster malware based on their memory and registers writing values, as well as an application of this technique to identify code reuse among malware samples from different families.

The thesis full text is available at `http://www.las.ic.unicamp.br/paulo/teses/20121128-PhD-Andre.Ricardo.Abed.Gregio-Malware.behavior.pdf`.

2.1. Behavioral Analysis

Existing taxonomies either address only one type of malware class [Weaver et al. 2003][Cooke et al. 2005][Dagon et al. 2007] [Boldt et al. 2004][Saroiu et al. 2004][Rutkowska 2006], or are closely tied to the standard classes and the current naming schemes [Filiol 2005][Karresand 2003]. Malicious programs behave most of the time similarly to benign programs. Therefore, to "analyze" a program, we need to pinpoint aspects of its behavior that serve to the purpose of characterizing malignity in it. Thus, we defined the **general behavior** and the **suspicious behavior** of a program. We consider the general behavior of a program as the set of actions—tuples "α_i" composed by source, operation (create, delete, write, terminate), object (file, process, network, registry, mutex, memory), and target—performed during its execution by an operating system.

The set of actions that compose a behavior can be divided into groups according to their nature: if an action interferes with the environment, i.e. changes the state of the system, it is part of an **active** subset of the behavior, otherwise, it is **passive**. There is also a subset of the general behavior that is **neutral**, i.e. the actions can be either active or passive, but they do not lead to a malign outcome. When a malicious program is executed, each of its actions can be considered suspicious, revealing important details about the infection. Hence, we define the suspicious behavior as follows:

Definition 1 *Let M_k be a sample whose general behavior $B(M_k)$ is divided into the active behavior $B_A(M_k)$ and the passive behavior $B_P(M_k)$. Then, $B(M_k) = B_A(M_k) \cup B_P(M_k)$. Let $B_N(M_k)$ be the malware's neutral behavior so that $B_N(M_k) \in B_A(M_k) \cup B_P(M_k)$. Thus, the suspicious behavior $B_S(M_k)$ is equal to $B_A(M_k) - B_N(M_k)$.*

Based on this, we proposed a naming scheme to identify potentially dangerous behaviors in unknown programs, producing the taxonomy shown in Table 1. Moreover, we evaluated over 12 thousand known malware samples collected from phishing e-mail messages, honeypots, public datasets and colleagues' cooperation. We submitted them to three antivirus engines in order to obtain their detection labels. At the time of scanning, $\approx 20\%$ of them were undetected by the selected AVs, $\approx 50\%$ were undecided, i.e., the AV did not agree with the detection label, and the remainder samples were labeled similarly. Then, the samples were submitted to dynamic analysis with our system, BehEMOT (Behavioral Evaluation of Malicious ObjecTs), which inspected for suspicious behaviors. This allowed us to perform the behavioral profiling of the monitored programs, as well as to identify these "unknown"—or "clean"–programs as suspicious (Figure 1).

Table 1. Proposed malware classes, suspicious behaviors and associated labels.

Class	Behavior	Label
Evader	Removal of Evidence	RemEvd [RE]
	Removal of Registries	RemReg [RR]
	AV Engine Termination	TerAVe [TA]
	Firewall Termination	TerFwl [TF]
	Notification of Updates Termination	TerUpd [TU]
	Language Checking	LngChk (Suspicious) [LC]
Disrupter	Scanning of Known-Vulnerable Service	VulScn [VS]
	E-mail Sending (Spam)	EmlSpm [ES]
	IRC/IM Known Port Connection	IrcPrt [IP]
	IRC/IM Unencrypted Commands	IrcCom [IC]
Modifier	Creation of New Binary	NewBin [NB]
	Modification of Existing System Binary	ChgBin [CB]
	Creation of Unusual Mutex	UnkMut [UM]
	Modification of the Name Resolution File	HstChg [HC]
	Modification of the Browser Proxy Settings	PacLdn [PL]
	Modification of the Browser Behavior	BhoInj [BI]
	Persistence	Persis [Pe]
	Download of Known Malware	DldKmw [DK]
	Download of Unknown File	DldUnk [DU]
	Driver Loading	DrvLdn [DL]
Stealer	Stealing of System/User Data	InfStl [IS]
	Stealing of Credentials or Financial Data	CrdStl [CS]
	System/user Information Reading	InfRdn (Suspicious) [IR]
	Process Hijacking	PrcHjk [PH]

We used our dynamic analysis system (BehEMOT) and knowledge about malware behavior to detect bankers, proposing BanDIT (Banker Detection and Infection Tracker). BanDIT employs a methodology composed of file system change identification, network traffic pattern matching, and image processing. Thus, we were able to identify whether a

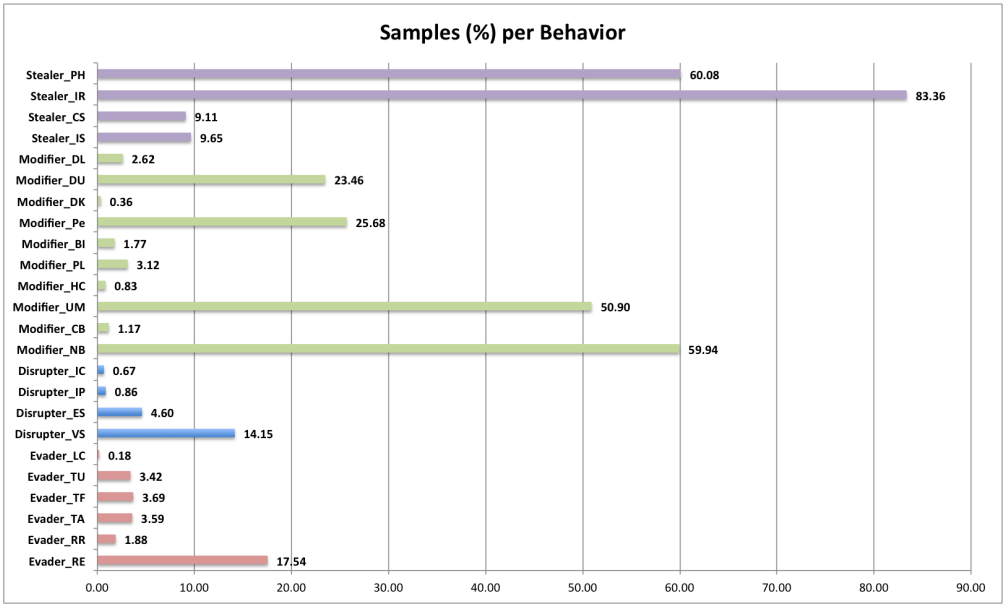


Figure 1. Behaviors observed in evaluated samples.

malware is a banker or not in 98.8% of cases, as well as to find IP and e-mail addresses involved in malware attacks.

We also used BehEMOT's output to search for visual similarities among execution behavior of malware samples assigned to the same family, as illustrated in Figure 2. Finally, we introduced another way for the extraction of behavioral profiles: to trace the malware execution using a debugger and to log arithmetic and logic instructions that modify values in memory or registers. To this end, we proposed a clustering algorithm able to group samples with a certain level of similarity, whose results may be used in the search of code reuse among malware.

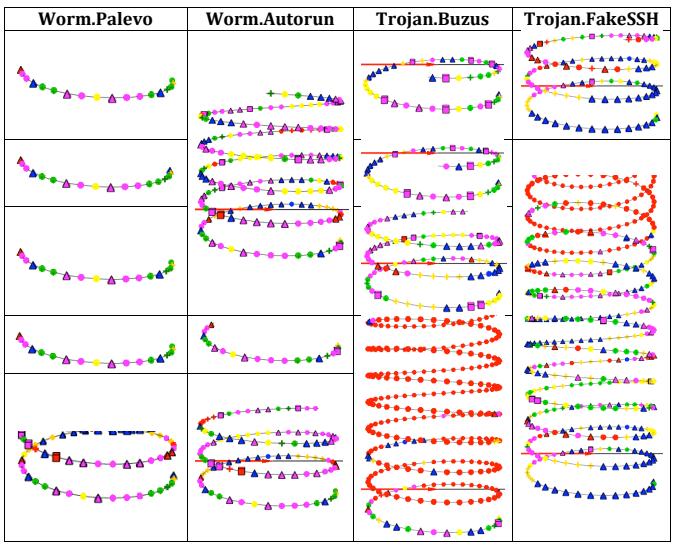


Figure 2. Visualization of execution behavior extracted from malware families.

3. Publications

The following list includes the published results related to this thesis (I assisted in the mentoring of the students whose papers I am not the first author). Most of them (international and national) are ranked by Brazilian Qualis Ranking 2012-2014.

1. **An Empirical Analysis of Malicious Internet Banking Software Behavior.** André Ricardo Abed Grégio, Vitor Monte Afonso, Victor Furuse Martins, Dario Simões Fernandes Filho, Paulo Lício de Geus, Mario Jino. ACM Symposium on Applied Computing (SAC). Coimbra, Portugal, March, 2013. **Qualis A1**.
2. **Tracking Memory Writes for Malware Classification and Code Reuse Identification.** André Ricardo Abed Grégio, Paulo Lício de Geus, Christopher Kruegel, Giovanni Vigna. 9^{th} Conf. on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), LNCS, Springer. Greece, July 2012. **Qualis B1**.
3. **Pinpointing Malicious Activities through Network and System-Level Malware Execution Behavior.** André Ricardo Abed Grégio, Vitor Monte Afonso, Dario Simões Fernandes Filho, Paulo Lício de Geus, Mario Jino, Rafael Duarte Coelho dos Santos. 12^{th} International Conference on Computational Science and Its Applications (ICCSA), LNCS, Springer Verlag. Brazil, June 2012. **Qualis B1**.
4. **Interactive, Visual-Aided Tools to Analyze Malware Behavior.** André Ricardo Abed Grégio, Alexandre Or Cansian Baruque, Vitor Monte Afonso, Dario Simões Fernandes Filho, Paulo Lício de Geus, Mario Jino, Rafael Duarte Coelho dos Santos. 12^{th} International Conference on Computational Science and Its Applications (ICCSA), LNCS, Springer Verlag. Brazil, June 2012. **Qualis B1**.
5. **A Hybrid Framework to Analyze Web and OS Malware.** Vitor Monte Afonso, Dario Simões Fernandes Filho, André Ricardo Abed Grégio, Paulo Lício de Geus, Mario Jino. IEEE International Conference on Communications (ICC), Proceedings of the IEEE ICC'12. Canada, June 2012. **Qualis A2**.
6. **A Malware Detection System Inspired on the Human Immune System.** Isabela Liane Oliveira, André Ricardo Abed Grégio, Adriano Mauro Cansian. 12^{th} International Conference on Computational Science and Its Applications (ICCSA), LNCS, Springer Verlag. Brazil, June 2012. **Qualis B1**.
7. **Behavioral analysis of malicious code through network traffic and system call monitoring.** André Ricardo Abed Grégio, Dario Simões Fernandes Filho, Vitor Monte Afonso, Rafael Duarte Coelho dos Santos, Mario Jino, Paulo Lício de Geus. Defense, Security and Sensing 2011, Proc. of SPIE. USA, April 2011.
8. **Visualization techniques for malware behavior analysis.** André Ricardo Abed Grégio, Rafael Duarte Coelho dos Santos. Defense, Security and Sensing 2011, Proceedings of SPIE. USA, April 2011.
9. **A hybrid system for analysis and detection of web-based client-side malicious code.** Vitor Monte Afonso, André Ricardo Abed Grégio, Dario Simões Fernandes Filho, Paulo Lício de Geus. IADIS International Conference WWW/Internet (ICWI'2011), Proceedings of ICWI, 2011. **Qualis B2**.
10. (In Portuguese) **Análise Visual de Comportamento de Código Malicioso.** Alexandre Or Cansian Baruque, André Ricardo Abed Grégio, Paulo Lício de Geus. Workshop de Trabalhos de Iniciação Científica e de Graduação (WTICG), Anais do XI SBSEG. Brazil, 2011.

11. (In Portuguese) **Sistema de coleta, análise e detecção de código malicioso baseado no sistema imunológico humano.** Isabela Liane Oliveira, André Ricardo Abed Grégio, Adriano Mauro Cansian. Conferência IADIS Ibero-Americana WWW/Internet (CIAWI), Anais da CIAWI, 2011.
12. (In Portuguese) **Análise Comportamental de Código Malicioso através da Monitoração de Chamadas de Sistema e Tráfego de Rede.** Dario Simões Fernandes Filho, André Ricardo Abed Grégio, Vitor Monte Afonso, Rafael Duarte Coelho dos Santos, Mario Jino, Paulo Lício de Geus. Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG), Anais do X SBSEG. Brazil, October 2010. **Qualis B4**.
13. (In Portuguese) **xFile: Uma Ferramenta Modular para Identificação de Packers em Executáveis do Microsoft Windows.** Victor Furuse Martins, André Ricardo Abed Grégio, Vitor Monte Afonso, Dario Simões Fernandes Filho, Paulo Lício de Geus. Workshop de Trabalhos de Iniciação Científica e de Graduação (WTICG), Anais do X SBSEG. Brazil, October 2010.
14. **Malware distributed collection and pre-classification system using honeypot technology.** André Ricardo Abed Grégio, Isabela Liane Oliveira, Rafael Duarte Coelho dos Santos, Adriano Mauro Cansian, Paulo Lício de Geus. Data Mining, Intrusion Detection, Information Security and Assurance, and Data Networks Security. Proceedings of SPIE Defense, Security and Sensing, USA, 2009.
15. (In Portuguese) [*Book Chapter*] **Técnicas para Análise Dinâmica de Malware.** Dario Simões Fernandes Filho, Vitor Monte Afonso, Victor Furuse Martins, André Ricardo Abed Grégio, Paulo Lício de Geus, Mario Jino, Rafael Duarte Coelho dos Santos. Minicursos do SBSEG 2011, pp.107–147, SBC, Brazil, 2011.

4. Conclusion

This document discussed several aspects related to the behavior of malicious programs, from the definition of potentially dangerous activities performed during an infection and the proposition of a behavior-based taxonomy, to the detection, clustering and visualization of malware. Our main objective is to provide a better understanding of how the diversity of current malware samples actually behave, as well as to aid in the development of practical and effective incident response procedures. To that extent, we discussed malware history and existing taxonomies, as well as introduced our own general and extensible taxonomy. Our proposal, yet of simple use and easy to understand, provides an overall view of malware infection. We evaluated the proposed behavior-centric taxonomy with over 12 thousand malware samples collected in the wild, showing that our analysis allows the identification of suspicious behaviors even in malware undetected by antiviruses. We also presented a system for Internet Banking malware detection, which was builti upon our dynamic analysis system. In addition, we leverage two interactive visualization tools that take advantage of behavioral profiles to aid in computer security incident response procedures. Furthermore, we introduce a novel way to classify malware with a good precision ($> 80\%$) that considered the values written in memory or registers.

References

Boldt, M., Carlsson, B., and Jacobsson, A. (2004). Exploring Spyware Effects. In *Nordic Workshop on Secure IT Systems (NORDSEC)*, Helsinki, Finland.

Cooke, E., Jahanian, F., and McPherson, D. (2005). The zombie roundup: understanding, detecting, and disrupting botnets. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, SRUTI'05, pages 39–44, Berkeley, CA, USA. USENIX Association.

Dagon, D., Gu, G., Lee, C. P., and Lee, W. (2007). A Taxonomy of Botnet Structures. In *23rd Annual Computer Security Applications Conference (ACSAC)*, pages 325–339.

Fallmann, H., Wondracek, G., and Platzer, C. (2010). Covertly probing underground economy marketplaces. In *Seventh Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*.

Filiol, E. (2005). *Computer viruses: from theory to applications*. Springer.

Holz, T., Engelberth, M., and Freiling, F. (2009). Learning more about the underground economy: a case-study of keyloggers and dropzones. In *Proceedings of the 14th European conference on Research in computer security*, ESORICS'09, pages 1–18.

Karresand, M. (2003). Separating Trojan Horses, Viruses and Worms - A Proposed Taxonomy of Software Weapons. In *IEEE Information Assurance Workshop*.

Lauinger, T., Kirda, E., and Michiardi, P. (2012). Paying for piracy? an analysis of one-click hosters' controversial reward schemes. In *15^{th} Internationa Symposium on Research in Attacks, Intrusions and Defenses (RAID)*.

Levchenko, K., Pitsillidis, A., Br, N. C., Halvorson, T., Kanich, C., Kreibich, C., Liu, H., Mccoy, D., Weaver, N., Paxson, V., Voelker, G. M., and Savage, S. (2011). Click trajectories: End-to-end analysis of the spam value chain. In *In Proceedings of IEEE Symposium on Security & Privacy*, pages 431–446.

Rutkowska, J. (2006). Introducing Stealth Malware Taxonomy. `http://invisiblethings.org/papers/malware-taxonomy.pdf`. Acesso realizado em 28 de fevereiro de 2012.

Saroiu, S., Gribble, S. D., and Levy, H. M. (2004). Measurement and Analysis of Spyware in a University Environment. In *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 141–153.

Stone-Gross, B., Abman, R., Kemmerer, R., Kruegel, C., Steigerwald, D., and Vigna, G. (2011a). The Underground Economy of Fake Antivirus Software. In *Proceedings of the Workshop on Economics of Information Security (WEIS)*.

Stone-Gross, B., Holz, T., Stringhini, G., and Vigna, G. (2011b). The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*.

Stringhini, G., Egele, M., Kruegel, C., and Vigna, G. (2012). Poultry markets: On the underground economy of twitter followers. In *Workshop on Online Social Network (WOSN)*. ACM.

Weaver, N., Paxson, V., Staniford, S., and Cunningham, R. (2003). A Taxonomy of Computer Worms. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM)*, pages 11–18, New York, NY, USA.

Índice dos autores

PROMOÇÃO:

ORGANIZAÇÃO:

PATROCÍNIO: