SBSeg16

XVI SIMPÓSIO BRASILEIRO
EM SEGURANÇA DA INFORMAÇÃO
E DE SISTEMAS COMPUTACIONAIS

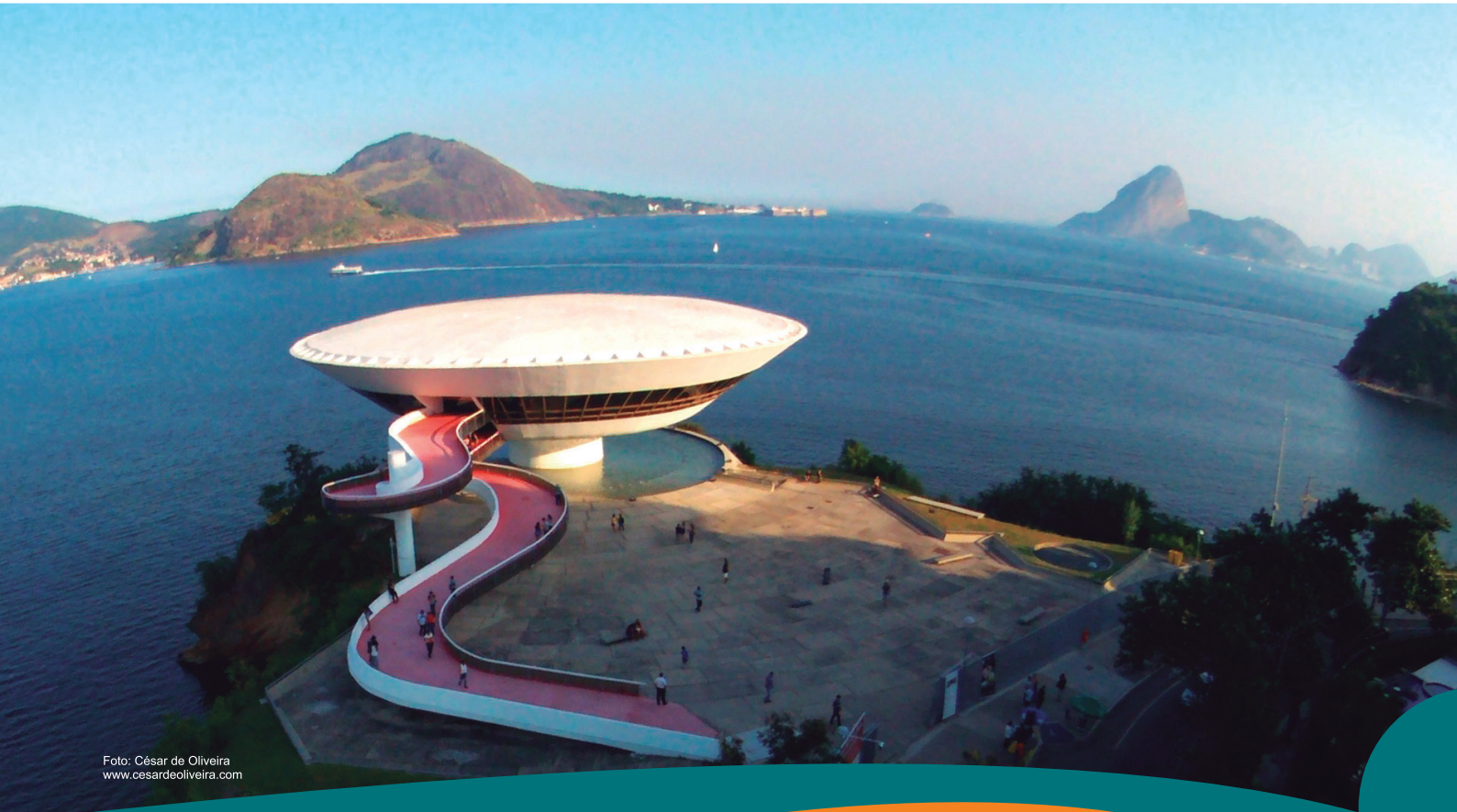


Foto: César de Oliveira
www.cesardeoliveira.com

MINICURSOS

SBSeg16

XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais

Niterói, RJ, 7 a 10 de novembro de 2016

MINICURSOS

Sociedade Brasileira de Computação – SBC

Organizadores
Igor Monteiro Moraes, UFF
Antônio Augusto de Aragão Rocha, UFF

Realização
Universidade Federal Fluminense – UFF
Sociedade Brasileira de Computação – SBC

Capa: Fatima Jane Ribeiro
Editoração: Ian Vilar Bastos, UFF

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da Universidade Federal Fluminense

S612 Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (16. : 2016 : Niterói, RJ)
Minicursos [do] XVI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 07 a 10 de novembro de 2016 / Sociedade Brasileira de Computação ; Organizadores, Igor Monteiro Moraes, Antônio Augusto de Aragão Rocha — Niterói, RJ: Sociedade Brasileira de Computação, 2016.
193 p.
CD-ROM : il. ; 4¾ pol..

ISBN: 978-85-7669-350-5

1. Ciência da computação. 2. Informática. 3. Segurança da informação. 4. Segurança de sistemas. I. Moraes, Igor Monteiro (org.) II. Rocha, Antônio Augusto de Aragão (org.) III. Título.

CDD 005.8 (21. ed)

Sociedade Brasileira de Computação – SBC

Contato
Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS

CNPJ: 29.532.264/0001-78
http://www.sbrc.org.br

Mensagem do Coordenador de Minicursos

Este livro apresenta a seleção de Minicursos da 16ª edição do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), realizado em Niterói-RJ, de 7 a 10 de novembro de 2016. Os minicursos representam uma oportunidade única para acadêmicos e profissionais para aprofundarem seus conhecimentos em temas relevantes e atuais da Área Segurança da Informação e de Sistemas Computacionais. A reconhecida qualidade dos textos produzidos pelos autores dos minicursos tem elevado estes textos à categoria de documentos de referência para trabalhos acadêmicos e formação complementar de estudantes, pesquisadores e profissionais.

Em 2016, 14 propostas de minicursos foram submetidas, um número expressivo que demonstra a importância deste evento no panorama nacional de pesquisa. Destas, 4 foram selecionadas para publicação e apresentação, representando assim uma taxa de aceitação de aproximadamente 29%. O comitê de avaliação dos minicursos foi composto por 11 renomados pesquisadores para a elaboração dos pareceres. Cada proposta recebeu ao menos 3 pareceres, gerando ao todo 54 revisões. Além disto, cerca de 90 mensagens foram trocadas entre os membros do comitê durante a fase de discussão.

Este livro reúne 4 capítulos produzidos pelos autores das propostas aceitas. O Capítulo 1 faz uma abordagem computational ao problema de proteção de privacidade, apresentando várias técnicas com suas primitivas criptográficas usadas para este fim. O Capítulo 2 apresenta a área de Computação Forense, com foco na área criminal, englobando alguns dos principais crimes cibernéticos e as técnicas e ferramentas usadas na área. O Capítulo 3 discute ataques e contramedidas em implementações em software de métodos criptográficos simétricos, e assimétricos baseados em curvas elípticas. Finalmente, o Capítulo 4 aborda os principais conceitos relacionados a *smart grid*, com foco nas vulnerabilidades e ataques que esse tipo de rede pode sofrer.

Como Coordenador de Minicursos, gostaria de expressar o meu agradecimento aos membros do Comitê de Programa por terem aceitado participar voluntariamente dessa empreitada e pelo excelente trabalho que fizeram no processo de avaliação e seleção dos minicursos. Gostaria de também agradecer aos coordenadores gerais do SBSeg 2016, Antônio Augusto de Aragão Rocha (UFF) e Igor Monteiro Moraes (UFF), pela disponibilidade e suporte providos ao longo de todo o processo e pela confiança depositada em mim para coordenar estes minicursos. Finalmente, gostaria de agradecer as autores por terem prestigiado este evento ao submeterem suas propostas de minicursos.

Michel Abdalla, ENS Paris & CNRS
Coordenador de Minicursos do SBSeg 2016

Comitê de Avaliação de Minicursos

Altair Santin, PUC-PR
Célio Vinicius Neves de Albuquerque, UFF
Eduardo Feitosa, UFAM
Leonardo Oliveira, UFMG
Luis Kowada, UFF
Luis Henrique Costa, UFRJ
Luiz Fernando Rust da Costa Carmo, Inmetro
Marcos Simplicio Jr - Escola Politécnica, USP
Michelle Wangham, Univali
Ricardo Dahab, UNICAMP
Rossana Andrade, UFC

Sumário

Capítulo 1

Introdução à Privacidade: Uma Abordagem Computacional

Fábio Borges
Laboratório Nacional de Computação Científica (LNCC)

Abstract

Information and communication technologies are continuously transforming how people interact in society. Many resources have been made available to simplify the lives of citizens. However, studies to ensure privacy have become increasingly needed. The leak of private information might have a serious impact on personal and professional life. The massive leak of private information might compromise the free will and democracy. The whole society can be manipulated. Considerable work has been done to protect privacy. This chapter aims to introduce techniques with their cryptographic primitives used to ensure privacy in various scenarios.

Resumo

Tecnologias da informação e comunicação estão continuamente transformando como as pessoas interagem na sociedade. Muitos recursos têm sido disponibilizados para simplificar a vida dos cidadãos. No entanto, estudos para garantir a privacidade têm se tornado cada vez mais necessários. O vazamento de uma informação privada pode causar sério impacto na vida pessoal e profissional. O vazamento massivo de informação privada pode comprometer o livre arbítrio e a democracia. Toda a sociedade pode ser manipulada. Muito tem sido feito para proteger a privacidade. Este capítulo visa introduzir técnicas com suas primitivas criptográficas usadas para garantir a privacidade em diversos cenários.

1.1. Introdução

Muitos estudantes que tiveram um curso de criptografia podem pensar que a privacidade depende apenas de uma decisão pessoal na qual ou se revela uma informação cifrada ou não se revela. No entanto, muitos problemas referentes à privacidade não são uma questão de escolha pessoal, mas são problemas computacionais complexos determinados por várias variáveis e oriundos de uma coletividade. Se por um lado, a garantia da privacidade está relacionada com problemas complexos, por outro lado o Direito à Privacidade é estabelecido no Artigo 12 da Declaração Universal dos Direitos Humanos (1948).

Este capítulo visa apresentar técnicas com suas primitivas criptográficas usadas para proteger a privacidade. Inicia-se apresentando diversos cenários nos quais a preservação da privacidade é fundamental. A violação do direito à privacidade em alguns destes cenários pode até mesmo comprometer a democracia. As seções subsequentes apresentam técnicas que são independentes de cenários específicos. Neste sentido, este texto é teórico, apesar de ter concretas motivações de cunho prático. Além disso, este capítulo apresenta uma coleta dos principais resultados sobre privacidade e deixa para os alunos o conhecimento e as referências necessárias para se aprofundarem em algum tópico específico. As primitivas criptográficas usadas nas técnicas para proteger a privacidade serão introduzidas antes de serem utilizadas. Logo, um aluno de graduação em computação pode seguir este texto sem a necessidade de ter cursado uma disciplina de criptografia, nem de segurança, como pré-requisito. Ao final deste capítulo, além de estar informado sobre os problemas de privacidade, os alunos de graduação terão habilidade de tratá-los de forma mais consciente e aplicar as técnicas mais adequadas às características de um problema de privacidade.

1.1.1. Principais Objetivos

Os principais objetivos deste capítulo são conscientizar o aluno da necessidade de preservar a privacidade em alguns cenários, introduzir métricas para avaliar as técnicas de proteção da privacidade para que o aluno possa avaliar melhor as técnicas e seus resultados, introduzir tecnologias usadas para proteger a privacidade junto com o conhecimento matemático necessário para se entender as técnicas simétricas e assimétricas usadas na proteção da privacidade, e como último objetivo, apresentar uma comparação das técnicas ensinadas neste texto.

1.1.2. Escopo e Estrutura do Texto

Este capítulo apresenta uma seleção de protocolos com seus algoritmos clássicos, *i.e.*, criptografia baseada em mecânica quântica está fora do escopo deste trabalho. Também não se deseja varrer todas as técnicas criptográficas usadas para proteger a privacidade, mas se objetiva varrer os protocolos mais importantes para proteger a privacidade nas aplicações mencionadas na Seção 1.2. Na sequência, a Seção 1.3 apresenta as métricas para a privacidade. A Seção 1.4 apresenta as técnicas simétricas e outras tecnologias para proteger a privacidade. A Seção 1.5 apresenta as técnicas de comprometimento junto com técnicas assimetrias. A Seção 1.7 apresenta comparações entres as principais técnicas. A seção 6 finaliza o texto com as conclusões. Para simplificar a leitura, o apêndice contém listas de acrônimos, abreviações e símbolos, além de um pequeno glossário.

1.2. O que é a privacidade?

Já sabemos que é um dos direitos humanos, mas sua definição é dependente da cultura de uma sociedade e do cenário a que se aplica à privacidade. Por exemplo, em certos países, os cidadãos escrevem os nomes dos moradores na caixa de correio, noutros não. Pois em outros, o nome na caixa do correio pode representar uma ameaça à segurança e privacidade. Salário é algo completamente privado em alguns países, noutros é público.

Nesta seção encontramos casos onde a privacidade se faz necessária na grande maioria dos países. Esta seção visa apresentar os problemas relativos à privacidade em vários cenários, a saber, votação eletrônica, sistemas de reputação, redes de sensores, cibermedicina, processamento de imagem, dinheiro eletrônico, sensoriamento móvel, computação em múltiplas partes, mundo acadêmico, e redes inteligentes. Esta não é uma lista completa de cenário. Na leitura dos cenários, o leitor pode imaginar muitos outros cenários. Finaliza-se a seção com uma breve introdução às leis que asseguram o direito à privacidade ao redor do mundo.

1.2.1. Votação Eletrônica

Fraudes em eleições podem comprometer a democracia. Por isto, faz-se necessário que os sistemas computacionais possam garantir a segurança das mensagens nos processos eleitorais. A violação do direito de os eleitores votarem secretamente pode coagi-los a votarem em candidatos que não votariam. Eleitores coagidos podem alterar o resultado da eleição o que também compromete a democracia. Por isto, faz-se necessário que os sistemas computacionais possam garantir a privacidade dos eleitores nos processos eleitorais.

Os sistemas de votação eletrônica devem ser similares ao processo tradicional de votação [Gritzalis 2002]. No entanto, nem todas as eleições baseadas em processos tradicionais são corretas, mas os processos de votação eletrônica deveriam ser corretos. De qualquer forma, o voto deve ser secreto. Além disto, os sistemas devem permitir verificações [Gritzalis 2002], *e.g.*, se o voto foi incluído na conta e se esta foi calculada corretamente. Verificações podem causar conflitos com privacidade. Para evitar conflito, pode-se imprimir o voto e depositá-lo em uma urna física. Urnas eletrônicas poderiam ser verificadas comparando seus resultados com os totais das urnas físicas. [Gritzalis 2002] apresenta os requisitos e princípios para sistemas de votação eletrônica que são resumidos na Tabela 1.1.

Sistemas de votação eletrônica podem ser construídos com algoritmos criptográficos, *e.g.*, [Cramer et al. 1997] apresentam um sistema de criptografia homomórfica baseado em [El Gamal 1985]. Saindo da computação clássica, [Vaccaro et al. 2007] apresentam sistemas de votação baseados em mecânica quântica.

1.2.2. Sistemas de Reputação

Sistemas de reputação, sistemas de recomendação e modelos de confiança descrevem esquemas para um novo usuário fazer escolhas mais assertivas baseadas em experiências de outros usuários. Sistemas de reputação podem ser baseados em algoritmos de ordenação como o PageRank. Sistemas de recomendação podem ser semelhantes a processos eleitorais, *e.g.*, onde clientes votam na sua empresa preferida. Modelos de confiança tentam

Tabela 1.1: Requisitos e princípios para sistemas de votação.

Requisitos	Princípios
Generalidade	Isomorfo ao tradicional Elegibilidade
Liberdade	Não-coercivo Nenhuma propaganda no entorno (sem *boca de urna*) Capacidade de votação não-válida (nulos, brancos, *etc.*)
Igualdade	Igualdade de candidatos Igualdade de eleitores Um eleitor - um voto
Sigilo	Voto secreto Segurança *v.* transparência
Direto	Votação não monitorada, mas contada
Democracia	Confiança e transparência Verificabilidade e prestação de contas Confiança e segurança Simplicidade

medir o quanto se pode confiar em uma empresa desconhecida baseando em evidências ou experiências relatadas por outros usuários. Todas estas três áreas de pesquisa têm uma grande intersecção. Em particular, a privacidade dos clientes testemunhas deve ser mantida para que os esquemas possibilitem que o novo cliente faça uma escolha assertiva. [Jøsang et al. 2007] desenvolveram uma interessante pesquisa literária sobre este tema.

Figura 1.1 esquematiza os clientes testemunhas, que já tiveram interação com uma empresa, provendo recomendações a um novo cliente que poderá tomar uma decisão mais assertiva devido a experiências relatadas. Certamente, o novo cliente tem que confiar nos clientes testemunhas.

[Kerschbaum 2009] desenvolve um sistema de reputação baseado em criptografia homomórfica e recomenda o uso da técnica de criptografia homomórfica desenvolvida por [Paillier 1999].

1.2.3. Redes de Sensores

Redes de sensores são redes formadas por sensores autônomos distribuídos espacialmente para coletar informações físicas e ambientais, *e.g.*, temperatura, umidade, ruído, *etc.*

Falhas na segurança podem gerar prejuízos financeiros enquanto o adversário se beneficia, *e.g.*, erros em previsões meteorológicas levam agricultores a plantar e colher em momentos errados. Logo, a colheita será reduzida e o custo do produto elevado no mercado. Semelhantemente, falhas na privacidade também podem ter implicações financeiras. Informações privilegiadas podem determinar melhor a precificação de imóveis. Em particular, um adversário pode saber se uma casa está vazia, se tem um habitante, e outras informações dos habitantes. Para isto, ele apenas precisa comparar as medições internas com as externas a residência. [Chan and Perrig 2003] apresentam um resumo dos problemas de segurança e privacidade em redes de sensores. [Peter et al. 2010] apresen-

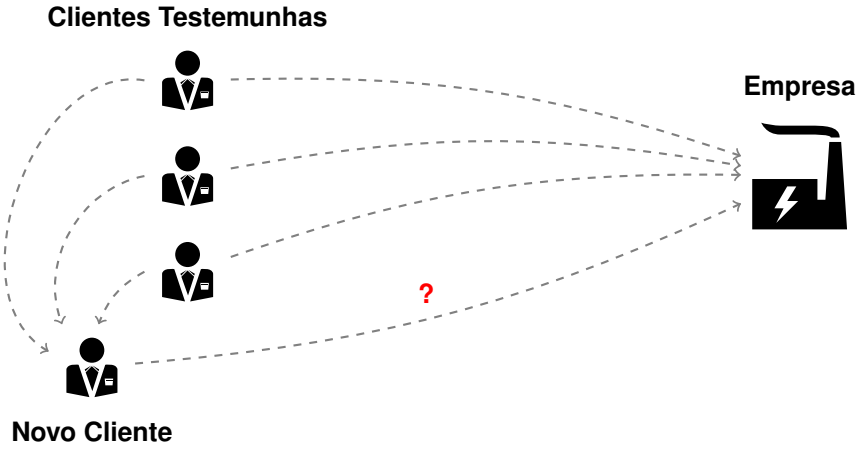


Figura 1.1: Cliente recebendo recomendações de outros clientes que já interagiram com uma empresa.

tam diversas técnicas de criptografia homomórfica para redes de sensores com especial atenção as baseadas em curvas elípticas.

Sensoriamento Móvel é um processo de coleta em dispositivos pessoais que são móveis formando um tipo especial de rede de sensores, e em geral, medem a posição geográfica de usuários de dispositivos móveis. [Li and Cao 2013] apresentam um esquema para agregar dados coletados, *e.g.*, a posição de vários usuários é agregada de forma que o agregador apenas descobre a quantidade de usuários em uma vizinhança.

1.2.4. Cibermedicina

Redes de sensores que coletam dados ligados a medicina também são sensíveis quanto a privacidade [Al Ameen et al. 2012]. Seguradoras e planos de saúde podem taxar clientes de forma diferenciada dependendo de dados coletados em tais redes. Um adversário poderia até mesmo saber dados sobre a emoção de um usuário de marca passos em determinados momentos.

Existem outros problemas de cibermedicina que não são relacionados a redes de sensores. Por exemplo, o prontuário eletrônico pode ser usado para rotular pessoas, e consequentemente, empresas poderiam se beneficiar ao violarem o sigilo médico. Em particular, dados de medicina do esporte poderiam privilegiar adversários.

[Bellare et al. 2007] apresentam como podemos fazer buscas em bases de dados criptografadas e [Naor and Shamir 1995] apresentam como podemos cifrar com várias chaves. Desta forma, poderíamos fazer pesquisas em prontuários sem ter acesso direto aos dados e poderíamos decifrar os prontuários com a chave do médico e do paciente, *i.e.*, quando ambos aprovassem o acesso.

1.2.5. Processamento de Imagem

[Zheng and Huang 2013] apresentam um esquema para proteção de imagens médicas baseado em criptografia homomórfica aditiva, *e.g.*, [Paillier 1999]. Além de imagens médicas, existem diversas aplicações de processamento de imagens que requerem cuidados

com a segurança e privacidade. Por exemplo, [Peter et al. 2013] apresentam um esquema para reconhecimento de faces que mantém a privacidade.

1.2.6. Dinheiro Eletrônico

As transações financeiras poderiam ser completamente eletrônicas. Entretanto, faz-se necessário manter a privacidade das transações. Caso contrário, cada pequena compra poderia ficar registrada para a vida toda. Com isto, um adversário poderia extrair informações de indivíduos e da sociedade que usar este recurso. Por exemplo, o adversário poderia saber sobre as escolhas, o comportamento, a personalidade, *etc.* Bitcoin é o dinheiro eletrônico mais conhecido. Uma análise de anonimato no uso de Bitcoins pode ser encontrada em [Reid and Harrigan 2013]. [Camenisch et al. 2007] apresentam um dos outros sistemas de dinheiro eletrônico. [Farhi et al. 2012] apresentam um sistema de dinheiro eletrônico baseado em mecânica quântica.

1.2.7. Computação em Múltiplas Partes

Computação em múltiplas partes é uma área de estudo que visa computar uma função em diversos dispositivos computacionais garantindo a segurança e privacidade dos dados. Em geral, a ideia é processar dados em nuvem de forma segura. Por exemplo, empresas de previsão meteorológica poderiam terceirizar o processamento sem preocupação de vazamento de informação. As buscas de novos medicamentos poderiam ser feitas em nuvem sem preocupações de vazamento de informação.

[Cramer et al. 2001] apresentam um esquema de computação em múltiplas partes baseada em um esquema de criptografia homomórfica aditiva criada por [Paillier 1999].

1.2.8. Privacidade no Mundo Acadêmico

Experimentos correlatos a seres humanos precisam passar por avaliações éticas. Além disto, existem outros problemas relacionados com privacidade. [Santini 2005] apresenta uma lista de dados confidenciais sobre revisões de importantes artigos científicos na área de computação que foram inicialmente rejeitados. Mais ainda, apresenta um link para um documento de avaliação de Albert Einstein com o nome de seu chefe que o avalia muito mal. Na academia, a privacidade é importante para não bloquear as pessoas de aprenderem por tentativa e erro. A privacidade dá o direito de avaliados e avaliadores errarem. Figura 1.2 ilustra que um revisor apresentou uma revisão classificando com nota contraria aos outros revisores gerando um ponto fora da curva, *i.e.*, um *outlier*. Os pontos em cinza no *boxplot* são as médias das avaliações. Certamente, os sistemas de avaliação devem classificar os melhores revisores e autores, mas isto deve ser feito na média dos últimos anos e não em pontos específicos.

Em instituições educacionais privadas onde se caracteriza uma relação de cliente e fornecedor, as leis dos direitos dos consumidores podem ser aplicadas.

> *Art. 43. O consumidor, sem prejuízo do disposto no art. 86, terá acesso às informações existentes em cadastros, fichas, registros e dados pessoais e de consumo arquivados sobre ele, bem como sobre as suas respectivas fontes.*
>
> *§1º Os cadastros e dados de consumidores devem ser objetivos, claros,*

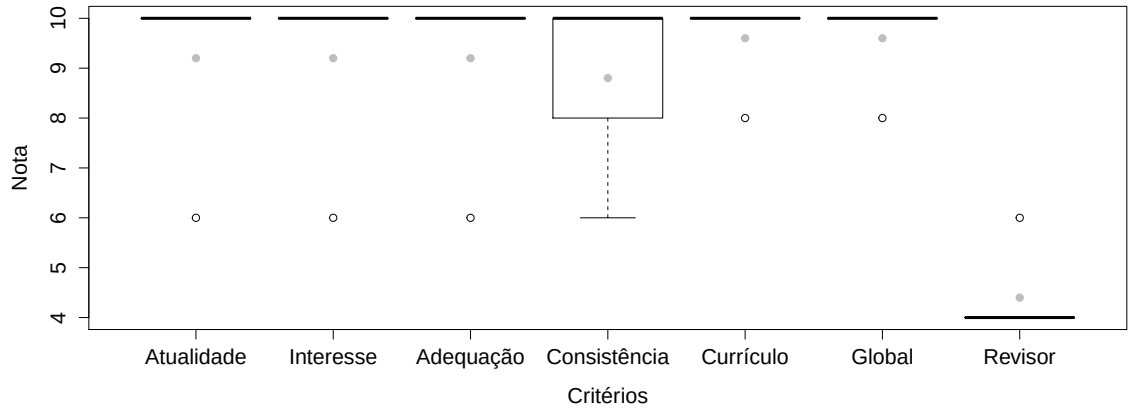


Figura 1.2: Nota obtida por critérios de avaliação.

> *verdadeiros e em linguagem de fácil compreensão, não podendo conter informações negativas referentes a período superior a* **cinco anos***.* [LEI Nº 8.078, DE 11 DE SETEMBRO DE 1990[1].]

> *Art. 59. Os órgãos públicos de defesa do consumidor devem providenciar a divulgação periódica dos cadastros atualizados de reclamações fundamentadas contra fornecedores.*
> *§3º Os cadastros deverão ser atualizados permanentemente, por meio das devidas anotações, não podendo conter informações negativas sobre fornecedores, referentes a período superior a* **cinco anos***, contado da data da intimação da decisão definitiva.* [DECRETO Nº 2.181, DE 20 DE MARÇO DE 1997[2].]

O tempo máximo de cinco anos deveria também ser usado na comunidade científica para retirada de informações sobre revisão de artigos científicos.

1.2.9. Redes Inteligentes (*Smart Grids*)

> *Smart grid é uma rede de pessoas, computadores, sensores em infraestruturas públicas que monitora e gerencia o uso de commodities.*
> [Borges de Oliveira 2017d]

Na maioria dos casos, os sensores são medidores inteligentes que enviam o consumo de eletricidade frequentemente para a companhia de energia. Neste caso, há diversos problemas de privacidade, *e.g.*, um adversário poderia detectar quando uma pessoa está em casa, se está sozinha, o que ela está assistindo na TV, que horas se levanta ou se deita, *etc.* [Borges de Oliveira 2017d]

[Borges de Oliveira 2017b] apresenta uma visão geral sobre redes inteligentes com os modelos de segurança e privacidade. [Borges de Oliveira 2017g] apresenta uma seleção de protocolos que protegem a privacidade dos consumidores.

[1] http://www.planalto.gov.br/ccivil_03/leis/L8078compilado.htm
[2] http://www.planalto.gov.br/ccivil_03/decreto/d2181.htm

Figura 1.3 esquematiza uma rede inteligente com os medidores inteligentes medindo o consumo da eletricidade e mandando os dados para suas respectivas companhias de distribuição de eletricidade. A figura apresenta duas redes, a saber, a rede de dados em azul e tracejado, e a rede elétrica em preto contínuo. Note que as medidas devem ser enviadas com frequência pela rede de dados as companhias, porque duas companhias estão virtualizando a rede de distribuição elétrica [Borges de Oliveira 2017f], *i.e.*, compartilhando uma rede de distribuição pública enquanto competem sem que uma terceira companhia controle o balanceamento das cargas elétricas. Logo, elas precisam prever o consumo com precisão para manterem equilíbrio da rede de distribuição elétrica. Para isso, precisam coletar e enviar os valores com máxima frequência, o que libera vários detalhes da vida privada dos clientes. Portanto, faz-se necessário o uso de protocolos que protejam a privacidade dos clientes.

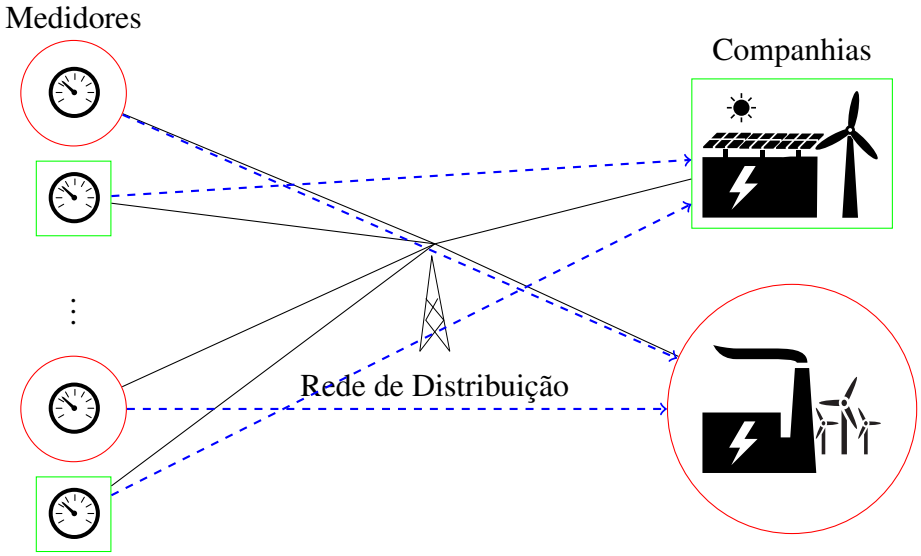


Figura 1.3: Rede elétrica e rede de dados em uma rede inteligente.

1.2.10. Leis sobre Privacidade

Como já vimos os diversos problemas referentes a privacidade, esta seção apresenta resumidamente de que forma as leis estão tratando a privacidade em diferentes culturas. Como já dito, a privacidade é um dos direitos humanos.

> *Ninguém deverá ser submetido a interferências arbitrárias na sua vida privada, família, domicílio ou correspondência, nem ataques à sua honra e reputação. Contra tais intromissões ou ataques todas as pessoas têm o direito à proteção da lei.*[3] [Artigo 12 da Declaração Universal dos Direitos Humanos (1948) [4]]

A legislação brasileira apresenta uma lei específica para privacidade na Internet.

[3]http://www.humanrights.com/pt/what-are-human-rights/videos/right-to-privacy.html
[4]http://www.un.org/en/universal-declaration-human-rights/

> *A garantia do direito à privacidade e à liberdade de expressão nas comunicações é condição para o pleno exercício do direito de acesso à internet.*
> [Art. 8o LEI Nº 12.965, DE 23 DE ABRIL DE 2014.]

O Brasil ainda não apresenta uma consolidação das leis sobre privacidade nos moldes das diretivas de proteção de dados Europeia[5]. No entanto podemos encontrar várias leis referentes a privacidade[6]. Os Estados Unidos da América têm uma regulamentação exacerbada em nível federal e estadual.

Figura 1.4 ilustra como os países são rigorosos quanto a lei de proteção de dados. O mapa é uma cópia de tela[7] gerada em setembro de 2016. No mesmo site, encontra-se um manual com a descrição de cada país.

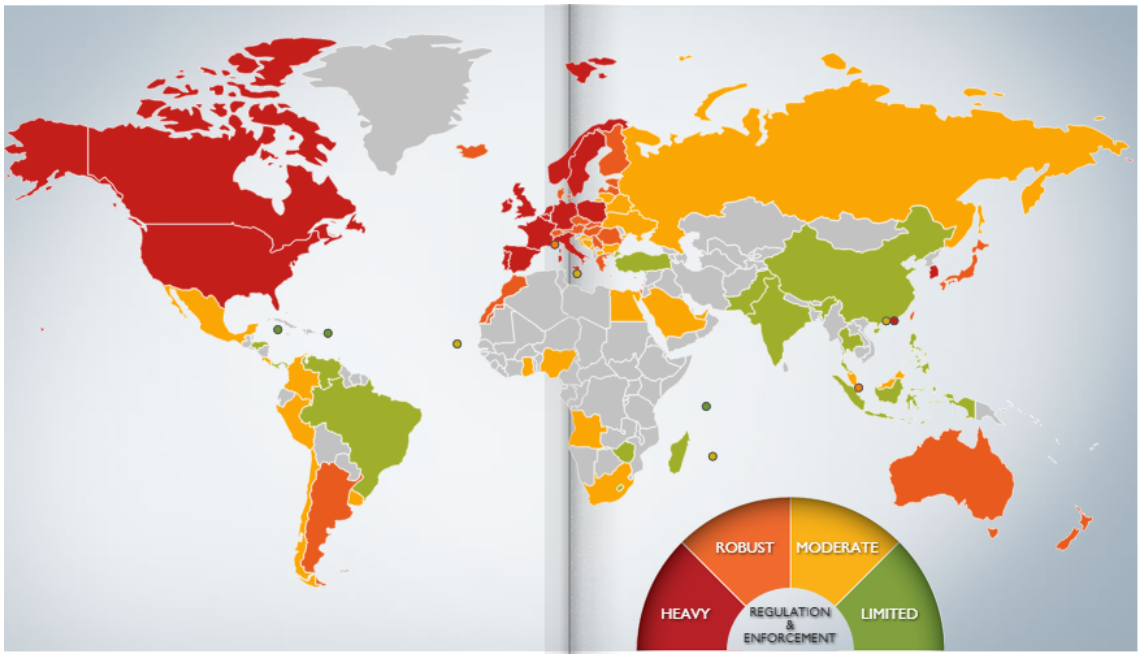


Figura 1.4: Mapa ilustrando como as leis de proteção de dados são tratadas no mundo.

1.3. Métricas para Privacidade

Privacidade diferencial [Dwork 2008], entropia [Díaz et al. 2003] e complexidade algorítmica são métodos usados para determinar se uma técnica criptográfica pode proteger a privacidade. Esta seção apresenta estes métodos e compara-os com uma abordagem introduzida recentemente e que se baseia em probabilidades [Borges de Oliveira 2016]. Assim como outros métodos, privacidade diferencial é usado para validar técnicas de proteção à privacidade, mas não estabelece uma métrica no sentido matemático. No entanto, podemos transformar os métodos para avaliar as técnicas em métricas matemáticas. Esta seção apresenta os métodos usados para avaliar as técnicas para proteger a privacidade. Também se apresenta como transformar estes métodos em métricas. Além disso, esta seção

[5]95/46/EC of 24 October 1995 e 2016/679 of 27 April 2016.
[6]http://dsic.planalto.gov.br/documentos/quadro_legislacao.htm
[7]https://www.dlapiperdataprotection.com/#handbook/world-map-section

apresenta as limitações das técnicas com suas primitivas criptográficas desenvolvidas para proteger a privacidade. Independentemente das técnicas usadas, os problemas de privacidade têm limitações intrinsecamente ligadas a várias variáveis [Borges de Oliveira 2016], por exemplo, se todos os outros participantes de um protocolo que usa uma técnica conspirarem, *i.e.*, conluiarem, os dados de um participante serão revelados independentemente do protocolo ou da técnica.

1.3.1. Anonimização

Além dos métodos já mencionados, temos também k-anonimato, l-diversidade, e t-proximidade [Li et al. 2007]. Todos usam o conceito de um conjunto E de dados indistinguíveis por um identificador em uma tabela. Estes métodos podem ser usados para anonimizar e avaliar a anonimização.

O método de k-anonimato suprime colunas de uma tabela ou troca-as por valores gerais de forma que cada E contenha pelo menos k registros. Este método apresenta sérias limitações visto que 4 pontos determinando a posição no tempo são suficientes para identificar de forma única 95% dos usuários de celular [De Montjoye et al. 2013].

Já l-diversidade requer que cada E tenha pelo menos l valores bem representados para cada coluna sensível. Bem representado pode ser definido de três formas:

1. pelo menos l distintos valores para cada coluna sensível;
2. para cada E, a entropia de Shannon é limitada, tal que $H(E) \geqslant \log_2 l$, onde

$$H(E) = -\sum_{s \in S} \Pr(E,s) \log_2(\Pr(E,s)),$$

   S é o domínio da coluna sensível, e $\Pr(E,s)$ é probabilidade da parte de linhas em E que tem valores sensíveis s;
3. o valor mais comum não pode aparecer com muita frequência e os incomuns não podem aparecer muito raramente.

Note que nem sempre as tabelas têm l distintos valores sensíveis. Além disto, a entropia da tabela tem que ser no mínimo $\log_2 l$. Finalmente, a frequência de valores comuns e incomuns normalmente não tendem a ser próximas.

Um E é dito ter t-proximidade, se a distância entre a distribuição de uma coluna sensível em E e a distribuição da coluna em toda a tabela não é mais do que um valor limiar t. Dizemos que uma tabela tem t-proximidade se todos E na tabela têm t-proximidade. Note que neste caso, t-proximidade gera uma relação inversa entre utilidade dos dados e privacidade.

1.3.2. Modelo de Segurança

Para garantir a privacidade, faz-se necessário garantir a segurança previamente. Certamente, o uso de primitivas criptográficas inseguras não garante a proteção da privacidade. Em particular, se os usuários acreditam que as mensagens estão seguras em um canal de comunicação, eles tendem a enviar mensagens confidenciais. Contrariamente, eles não

enviam mensagens confidenciais se o canal for inseguro. Portanto, a falsa sensação de segurança representa uma ameaça do maior que a transmissão sem proteção.

São diversos os exemplos de falsa sensação de segurança. Um decreto[8] vigente desde 2009 recomenda o uso de um algoritmo de hash chamado MD5. No entanto, o algoritmo já era considerado inseguro desde 2005 [Wang and Yu 2005].

Imagine que uma bolsa de valores ou uma companhia de cartão de crédito envie as transações financeiras por e-mail em um arquivo PDF criptografado com algoritmos seguros. Entretanto, os clientes não podem escolher suas senhas, mas a instituição escolhe as senhas baseadas em sequências numéricas. Esta informação é conhecida por todo cliente. Logo, um adversário poderia saber disto facilmente, e executando o Código 1.1, ele poderia descobrir a senha de um cliente em menos de um minuto. Interceptando os e-mails, o adversário poderia saber todas as transações de todos os clientes enquanto que eles estariam se sentindo seguros porque ninguém consegue quebrar o AES.

Código 1.1 Script em Bash de Força Bruta

```
1: #!/bin/bash
2: #clear
3: echo
4: echo "Processando...␣"
5: for i in {0..999999}
6: do
7:         pdfinfo -upw $i secreto.pdf  &> /dev/null
8:         if [ "$?" -eq "0" ]; then
9:                 echo $i > senha.out
10:                break
11:        fi
12: done
13: cat senha.out
14: xpdf -upw $i secreto.pdf &
```

O modelo de segurança criado por Shannon considera que tanto a geração quanto distribuição das chaves são seguras. O adversário conhece os algoritmos usados na criptografia e tem acesso ao canal de comunicação. Ele apenas não sabe quais são as chaves seguras. Figura 1.5 esquematiza o modelo de segurança criado por Shannon. O adversário tem acesso às mensagens encriptadas $\mathfrak{M}_{i,j}$ pelo usuário i no tempo j, mas não tem acesso às mensagens $m_{i,j}$ decifradas. Ele também conhece a função que encripta Enc e a função que decripta Dec, mas é enfatizado que ele não tem acesso às chaves que devem ser seguras e nem às mensagens $m_{i,j}$.

Note que o modelo foi criado antes do advento da criptografia assimétrica. Porém, é válido para algoritmos de criptografia simétricas e assimétricas.

1.3.3. Modelo de Privacidade

O modelo de segurança não protege a privacidade porque o adversário tem a mesma informação que o destinatário tem. Pode-se, até mesmo considerar que o destinatário é o

[8]http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/D6870.htm

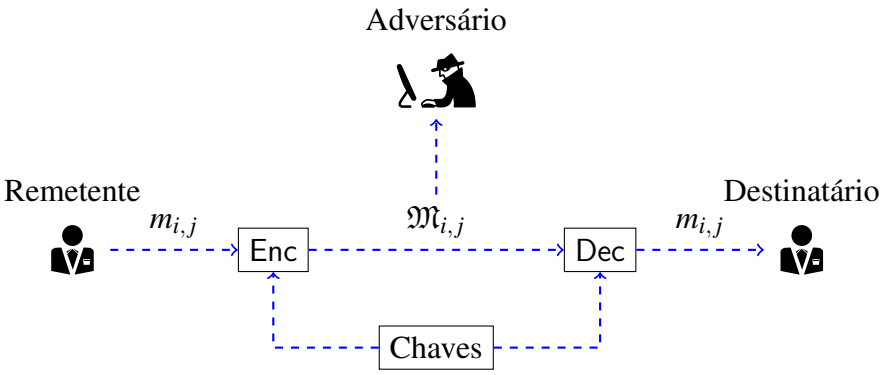


Figura 1.5: Modelo de Shannon para a segurança das mensagens $m_{i,j}$.

adversário. Consequentemente, se o adversário sabe que apenas o usuário i enviou uma mensagem $m_{i,j}$ no tempo j, então a privacidade foi invadida. Mesmo que apenas um usuário i tenha uma mensagem $m_{i,j}$ para ser enviada, faz-se necessário que vários usuários enviem mensagens para que o adversário não identifique o remetente. Se a mensagem $m_{i,j}$ fosse um texto, todos os outros usuários enviariam um texto em branco. Neste caso, o destinatário recebe uma consolidação encriptada $\mathfrak{C}_j$ em vez de receber uma mensagem $m_{i,j}$. Consequentemente, ele pode decriptar a consolidação encriptada $\mathfrak{C}_j$ e ter acesso à consolidação c_j das mensagens no tempo j. Logo, precisamos de um conjunto de usuários i e dizemos que o conjunto tem cardinalidade I, *i.e.*, o número de usuário é número de usuários. Cada consolidação c_j acontece em um tempo j e a cardinalidade do conjunto de consolidações é definida pelo número de tempo J. Certamente, o número de tempo J não tem que ser limitado. Apenas fixamos um número inteiro para podermos definir interessantes momentos em um protocolo. Por exemplo, o tempo j pode ser o ano que ocorreu uma eleição e o número de tempo J pode propiciar uma avaliação das últimas J eleições. De forma mais prática e geral, se a mensagem $m_{i,j}$ é uma transação do usuário i no tempo j, então o número de tempo J pode representar o total de transações em um mês. Normalmente, a consolidação c_j em um tempo j é chamada de agregação e se trata de uma soma. No entanto, elas podem ter significados diferentes. Em redes inteligentes, podemos ter cenários mais complexos e diferenciar a medição agregada da medição consolidada. Neste caso, agregação é uma única medição que engloba vários clientes que moram em uma área, normalmente medida com um *phasor measurement unit* (PMU) em transformadores ou sustações elétricas. Consolidação é a junção de medições individuais dos mesmos clientes um uma única informação, normalmente as medições dos clientes são feitas com medidores inteligentes em suas residências.

Figura 1.6 esquematiza a consolidação de mensagem $m_{i,j}$ de um número de usuários I em um tempo j. O adversário tem todas as informações do destinatário, mas não dos remetentes. Dependendo do protocolo de proteção da privacidade, o adversário pode ou não ter acesso às mensagens encriptadas $\mathfrak{M}_{i,j}$ durante o processo de consolidação. O adversário tem acesso à consolidação encriptada $\mathfrak{C}_j$, mas tal conhecimento não é tão interessante, pois o adversário tem acesso à consolidação c_j, que não tem criptografia.

Note que a ideia central por trás dos métodos de anonimização é a consolidação. Muitas aplicações têm usado alguma primitiva de criptografia homomórfica aditiva

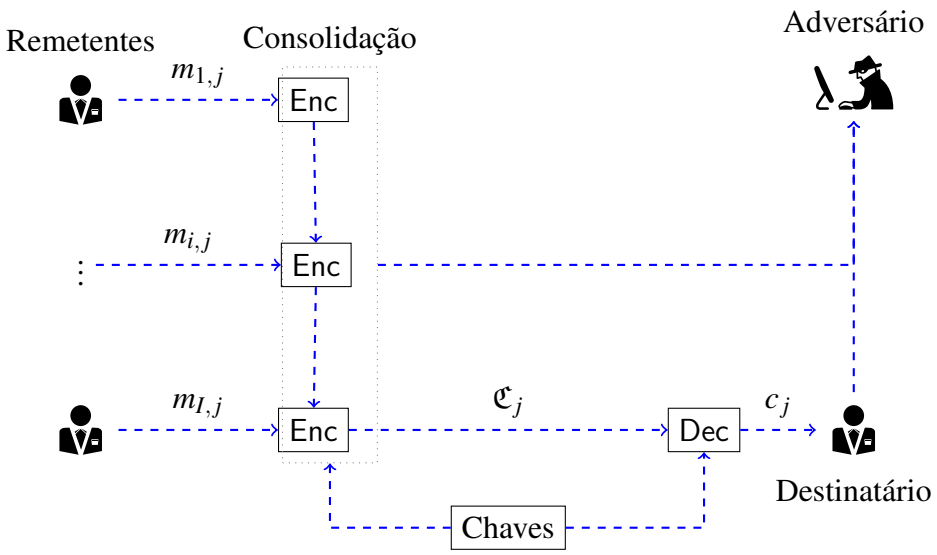


Figura 1.6: Modelo de consolidação das mensagens para preservar a privacidade.

(PCHA). Dizemos que a técnica de criptográfica é PCHA se ela permite descriptografar uma única mensagem $m_{i,j}$ e permite somar mensagens, *i.e.*,

$$\prod_{i=1}^{I} \mathsf{Enc}\left(m_{i,j}\right) = \mathsf{Enc}\left(\sum_{i=1}^{I} m_{i,j}\right) = \mathsf{Enc}\left(c_j\right) = \mathfrak{C}_j.$$

Tais funções devem ser probabilísticas, *i.e.*, mensagens iguais devem resultar mensagens encriptadas diferentes. Se não, o adversário pode identificar mensagens em branco pela sua frequência. Portanto, ele poderia identificar quem mandou ou não uma mensagem.

1.3.4. Definição de Métrica

Métrica é uma função que mede a distância entre dois pontos. Existem vários tipos de métricas. Elas podem nos ajudar a comparar algoritmos e medir a segurança e privacidade.

Primeiramente, vamos nos lembrar como a métrica é definida. Dado um conjunto $\mathscr{C}$ e uma função $d : \mathscr{C} \times \mathscr{C} \to \mathbb{R}^+$, onde $\mathbb{R}^+$ representa o conjunto dos números reais não-negativos, dizemos que d é uma métrica se as seguintes condições são satisfeitas para todo $x, y, z \in \mathscr{C}$:

1. $d(x,y) \geqslant 0$ positivamente definida
2. $d(x,y) = 0 \Leftrightarrow x = y$ identidade
3. $d(x,y) = d(y,x)$ simetria
4. $d(x,z) \leq d(x,y) + d(y,z)$ desigualdade triangular

Métricas podem ser aplicadas a base de dados, sequências de dados, redes com nós e probabilidades associadas aos nós, séries temporais, *etc.* Os diferentes métodos para medir a privacidade foram construídos para cenários diferentes. É possível transferir os

métodos em diversos cenários. Certamente, todas as informações necessárias ao método têm que estar disponíveis. Na sequência, a nomenclatura das aplicações iniciais usadas em cada método é mantida.

1.3.5. Privacidade Diferencial

A noção de privacidade diferencial é semelhante a noção de indistinguibilidade em criptografia. Para defini-la, precisamos de um número real positivo ε e um algoritmo probabilístico $\mathscr{A}$ que pega uma base de dados como entrada. O algoritmo $\mathscr{A}$ é privado ε-diferencialmente se para toda base de dados D_1 e D_2 que difere em um único elemento, e para todos os subconjuntos de S da imagem de $\mathscr{A}$, temos

$$\Pr[\mathscr{A}(D_1) \in S] \leq e^{\varepsilon} \times \Pr[\mathscr{A}(D_2) \in S],$$

onde a probabilidade é controlada pela aleatoriedade usada pelo algoritmo.

Note que privacidade diferencial não é uma métrica no sentido matemático. Porém, se os algoritmos manterem a probabilidade conforme a entrada, podemos construir uma métrica para comparar a distância entre dois algoritmos calculando

$$d(\mathscr{A}_1, \mathscr{A}_2) = |\varepsilon_1 - \varepsilon_2|.$$

Assim, podemos determinar se dois algoritmos são equivalentes $\varepsilon_1 = \varepsilon_2$, ou ainda, podemos determinar a distância a um algoritmo ideal

$$d(\mathscr{A}_1, \mathscr{A}_{\text{ideal}}) = |\varepsilon_1 - 0|.$$

1.3.6. Entropia

O grau de anonimato g pode ser medido com a entropia de Shannon

$$H(X) = \sum_{i=1}^{N} \left[p_i \cdot \log_2 \left(\frac{1}{p_i} \right) \right],$$

onde $H(X)$ é a entropia da rede, N é o número de nós e p_i é a probabilidade associada a cada nó i. A entropia máxima ocorre quando temos uma probabilidade uniforme, *i.e.*, todos os nós sãs equiprováveis $1/N$, o que gera

$$H_M = \log_2(N).$$

Logo, o grau de anonimato g é definido por

$$g = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M}.$$

Similarmente, podemos construir uma métrica para comparar a distância entre duas redes calculando

$$d(g_1, g_2) = |g_1 - g_2|.$$

Analogamente, podemos determinar se elas são equivalentes $g_1 = g_2$. Consequentemente, podemos determinar a distância a uma rede de anonimato ideal

$$d(g_1, g_{\text{ideal}}) = |g_1 - 1|.$$

A rede pode ser trocada por um banco de dados, mas neste modelo, cada registro tem que ter uma probabilidade associada a ele.

1.3.7. Complexidade

Análise de complexidade também pode se transformar em uma métrica para medir o tempo necessário para quebrar um algoritmo criptográfico que protege a privacidade. A medida pode ser de forma assintótica ou em número de passos.

De forma geral, mesmo que a complexidade impeça a quebra do algoritmo devido a um problema matemático, mesmo que o algoritmo nos forneça uma privacidade diferencial ideal, e mesmo que o grau de anonimato seja máximo, em todas estas situações a privacidade pode ser violada. Por exemplo, considere uma votação com 4 eleitores, se 3 conspirarem, a privacidade do quarto será violada independentemente de qualquer algoritmo ou protocolo. [Borges de Oliveira 2017e] apresenta como quebrar protocolos baseados em ruído que tem privacidade diferencial para redes inteligentes.

O algoritmo criptográfico deve garantir a privacidade da mesma forma que garante segurança. Uma mensagem cifrada deve ter medidas de confidencialidade máximas para a privacidade, assim como tem para a criptografia. Desta forma, devemos usar a complexidade do melhor algoritmo que resolve um problema matemático associado ao algoritmo criptográfico. Logo, a complexidade pode ser usada para determinar o atual nível de segurança e privacidade de uma mensagem cifrada. Além disto, devemos considerar os ataques a privacidade que são independentes do algoritmo criptográfico.

1.3.8. Contagem e Probabilidades

Esta seção aborda casos de ataques independentes dos algoritmos escolhidos. Logo, temos que contar quantas opções seriam possíveis e quais suas probabilidades.

Voltando ao exemplo da votação, suponha que os 4 eleitores votaram em 3 sins e 1 não. Neste caso, um adversário sabe que a probabilidade de um eleitor ter votado sim é $3/4$ e de $1/4$ para não. Se tivéssemos mais eleitores e o resultado fosse 15 sins e 5 nãos, então as respectivas probabilidades também seriam $3/4$ e $1/4$. A eleição é mais fácil de contar porque o voto é binário, ou eu voto ou não em um candidato. A mesma lógica se aplica ao caso de vários candidatos.

Diferente dos casos binários, pode-se desejar manter a privacidade de valores oriundos de medidas. Para um adversário descobrir uma série temporal de três pontos, ele pode representar cada ponto por um número de estrelas, *i.e.*, o total de símbolos $\star$. Assim, o adversário pode separar o total de estrelas em três caixas. Suponha que a soma da série seja 7, então uma possibilidade seria $\boxed{\star\star\star\star}\,\boxed{\star}\,\boxed{\star\star}$. Por simplicidade, o adversário pode separar as estrelas por barras em vez de caixas. Logo, $\star\star\star\star\ |\star|\star\star$ tem a mesma solução. Com esta notação, a combinação de 7 estrelas mais 2 barras escolhidas 7 estrelas determina o número possível de soluções, matematicamente temos

$$\binom{7+2}{7} = \frac{9!}{7!(9-7)!} = 36.$$

De forma geral, se t é o número total de pontos da série temporal e se s sua soma, então o número de possíveis séries temporais para o adversário decidir a correta é

determinado por s mais $t-1$ escolhendo s, portanto

$$\binom{s+t-1}{s} = \frac{(s+t-1)!}{(t-1)!s!} = \binom{s+t-1}{i-1}. \quad (1)$$

Múltiplas séries temporais podem formar uma tabela, *e.g.*, a lista de candidatos com votos por estados. Para evitar que o candidato eleito privilegie estados, o tribunal eleitoral poderia divulgar apenas o número de votantes por estado e o total de votos por candidato. No entanto, os candidatos poderiam inferir os possíveis votos por estado [Borges de Oliveira 2017e] e dados de eleições passadas poderiam ajudar. Note que tais somas poderiam ser computadas com dados criptografados de forma muito mais robusta que anonimização por k-anonimato, l-diversidade e t-proximidade. Mesmo assim, dependendo do tamanho da tabela e de seus valores, os valores podem ser encontrados.

Em geral, podemos abstrair os votos e outros valores para uma medição. Desta forma, técnicas de anonimização tentam reduzir o número de medições em tabelas. Contrariamente a intuição, quanto menor o número de medições, maior a chance de descobrilas [Borges de Oliveira 2017e].

1.4. Técnicas Simétricas e outras Tecnologias

Além de diversas técnicas que usam criptografia, diversas tecnologias têm sido desenvolvidas para proteger a privacidade, por exemplo, o uso de baterias e outros buffers em smart grids. Esta seção apresenta técnicas baseadas em ruído, pseudônimo e redes de anonimato, e DC-Net simétrica (SDC-Net) introduzidas por [Chaum 1988] cujo nome vem de *Dining Cryptographers Network* (DC-Net). Antes de cada técnica, apresenta-se o fundamento matemático necessário para entender a técnica. Apresenta-se também as limitações das técnicas e tecnologias. Um problema das técnicas simétricas é o número de chaves necessárias para troca de mensagens. Especificamente, o número de chaves cresce quadraticamente com o número de participantes. Existem tecnologias viáveis e inviáveis. O uso de caixas de água em residências para proteger a privacidade em uma smart grid que controla o fluxo de água é viável. Já o uso de baterias para proteger a privacidade em uma smart grid que controla o fluxo da energia elétrica é financeiramente inviável. Técnicas que usam criptografia são muito mais acessíveis. Em particular, SDC-Net podem garantir uma segurança incondicional, mas neste caso, não são adequadas para aplicações reais devido a limitação na qual a chave criptográfica somente pode ser usada uma vez.

1.4.1. Ruído

Várias técnicas de anonimização usam uma função pseudoaleatória ou uma distribuição gaussiana ou laplaciana para introduzir um ruído nas mensagens dos usuários. Em geral, sabe-se para onde a soma dos ruídos convergem. [Borges de Oliveira 2017h] mostrou que tais técnicas não funcionam para smart grids. Tais técnicas não funcionam porque um adversário poderia usar várias medidas com ruído de um determinado horário e a soma das medidas com ruído convergem para soma sem ruído. Logo, um adversário poderia criar um perfil do cliente.

Em geral, um adversário pode descobrir uma mensagem encriptada $\mathfrak{M}_{i,j}$ para qualquer aplicação de ruído onde se saiba sua convergência. Suponha que $\mathfrak{M}_{i,j} = m_{i,j} +$

$r_{i,j}$ é a mensagem $m_{i,j}$ com ruído $r_{i,j}$ de usuário i no tempo j. Por simplicidade, suponha que a distribuição tenha valor esperado $\mu = 0$, logo a série de ruídos converge para zero

$$\sum_{i}^{\infty} r_{i,j} = \sum_{j}^{\infty} r_{i,j} \to 0.$$

De fato, a série aproxima de zero caso ela seja truncada com um número suficiente de termos. Desta forma, um adversário gera uma série de ruídos r'_l com a mesma distribuição e computa

$$\mathfrak{M}_{i,j} + \sum_{l=1}^{L} r'_l \to m_{i,j},$$

onde L é um número suficientemente grande.

1.4.2. Pseudônimo e Redes de Anonimato

Em vez de usar a verdadeira identidade, usar um pseudônimo para cada serviço é uma forma de manter anonimato. No entanto, um adversário pode rastrear o meio de comunicação e identificar um usuário pela origem da conexão. Por isto, faz-se necessário que a conexão venha de uma rede de anonimato, *i.e.*, um adversário não consegue rastrear a origem da conexão.

Redes de anonimato podem ser construídas de forma física, *e.g.*, todas as máquinas conectadas no mesmo barramento recebendo todas as mensagens. Assim, o adversário apenas sabe que o pseudônimo é usado em uma daquelas máquinas. Além disto, redes de anonimato podem ser construídas por software. [Chaum 1981] introduziu o conceito de Mix networks que são redes de anonimato construídas por servidores proxy chamados de mixes.

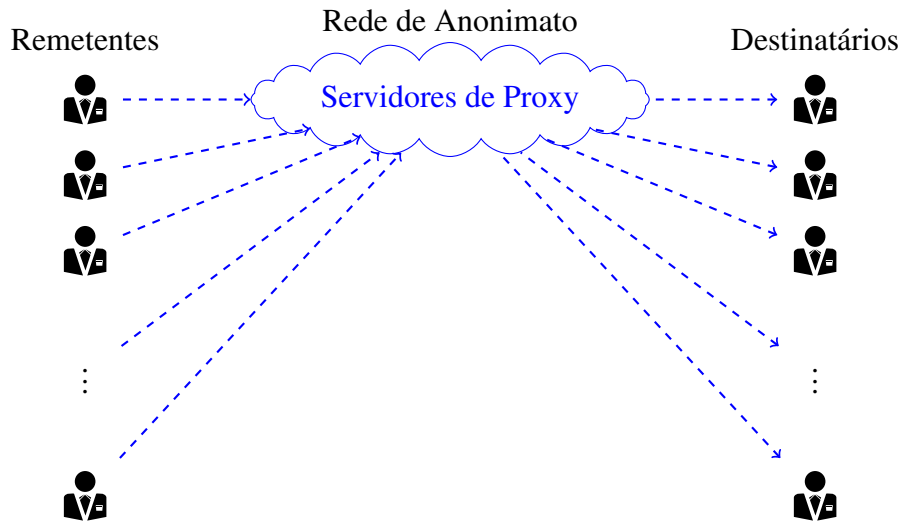


Figura 1.7: Uma mix network com possíveis remetentes e possíveis destinatários.

[Borges et al. 2012] apresentam uma relação inversa entre performance e privacidade em redes de anonimato que usam pseudônimos. Desta forma, funções criptográficas homomórficas são mais eficientes que usar redes de anonimato e pseudônimos. Tais

funções permitem manipular mensagens criptografadas, *i.e.*, podemos aplicar operações matemáticas e fazer buscas em dados cifrados. Redes de anonimato podem ser construídas sem a necessidade de terceiros como servidores de proxy. SDC-Net podem ser usadas como redes de anonimato.

1.4.3. DC-Nets Simétricas

[Chaum 1988] introduziu o conceito de SDC-Net através de um problema de privacidade com sua respectiva solução. Resumidamente, três criptógrafos foram pagar a conta do jantar, mas a conta já estava paga. Eles queriam saber se alguma agência de segurança pagou ou se um deles pagou, mas sem revelar a identidade de quem pagou. Primeiramente eles combinam uma senha com os outros dois. Figura 1.8 esquematiza a distribuição de chaves para SDC-Net.

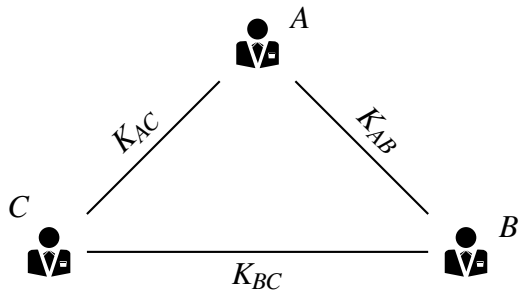


Figura 1.8: Distribuição de chaves em SDC-Net.

Como a porta lógica XOR $\oplus$ cancela sequências igual de bits, *e.g.*, $101101 \oplus 101101 = 000000$. Assim, $k \oplus 0 = k$ e $k \oplus 1 \oplus k = 1$ para todo k. Além disto, $K_{AB} \oplus K_{AC}$ é conhecido apenas pelo A. Sem perda de generalidade, suponha que A pagou. Logo, A revela $K_{AB} \oplus K_{AC} \oplus 1$, enquanto B revela $K_{AB} \oplus K_{BC}$, e C revela $K_{AC} \oplus K_{BC}$. Todos podem calcular

$$K_{AB} \oplus K_{AC} \oplus 1 \oplus K_{AB} \oplus K_{BC} \oplus K_{AC} \oplus K_{BC} = 1.$$

Caso ninguém tivesse pago a conta seria

$$K_{AB} \oplus K_{AC} \oplus K_{AB} \oplus K_{BC} \oplus K_{AC} \oplus K_{BC} = 0.$$

Sem perda de generalidade, suponha agora que o usuário A do protocolo é um adversário e calcula

$$K_{AB} \oplus \underbrace{(K_{AB} \oplus K_{BC})}_{\text{revelado por } B} = K_{BC}$$

e

$$K_{Ac} \oplus \underbrace{(K_{AC} \oplus K_{BC})}_{\text{revelado por } C} = K_{BC}.$$

Portanto, a chave não pode ser usada novamente.

Se A não tivesse enviado a mensagem que pagou, ou A teria obtido

$$K_{AB} \oplus \underbrace{(K_{AB} \oplus K_{BC} \oplus 1)}_{\text{revelado por } B} = K_{BC} \oplus 1$$

e

$$K_{Ac} \oplus \underbrace{(K_{AC} \oplus K_{BC})}_{\text{revelado por } C} = K_{BC},$$

ou A teria obtido

$$K_{AB} \oplus \underbrace{(K_{AB} \oplus K_{BC})}_{\text{revelado por } B} = K_{BC}$$

e

$$K_{Ac} \oplus \underbrace{(K_{AC} \oplus K_{BC} \oplus 1)}_{\text{revelado por } C} = K_{BC} \oplus 1.$$

Note que A não tem uma informação sobre quem pagou, que já não tinha anteriormente. Se as senhas forem verdadeiramente aleatória e forem usadas apenas uma vez, então temos um segredo perfeito para privacidade.

Para evitar que a chave K_{BC} acordada entre B e C seja revelada, eles podem usar uma função de hash H com um tempo j para proteger as chaves em vez deles usarem diretamente as chaves. A chave pode ser concatenada junto com o tempo j, e então, calculada a função de hash H, *e.g.*, $\mathsf{H}(K_{AB}||j)$ em vez de apenas K_{AB}, o símbolo $||$ significa concatenação. Desta forma, o mesmo ataque revelaria apenas $\mathsf{H}(K_{BC}||j)$. Como o tempo j não se repete, a função de hash geraria a aleatoriedade necessária para proteger a chave.

Em vez de usarmos uma senha para cada par de usuários, cada usuário pode mandar uma senha para cada um dos outros. No exemplo prévio, as conexões entre os usuários formam um grafo, mas agora forma um grafo orientado. Não necessariamente, todos os usuários teriam que estabelecer chaves com todos os outros. Usuários poderiam ter chaves apenas com os usuários de sua confiança. Figura 1.9 esquematiza uma SDC-Net formando um dígrafo completo, *i.e.*, orientado.

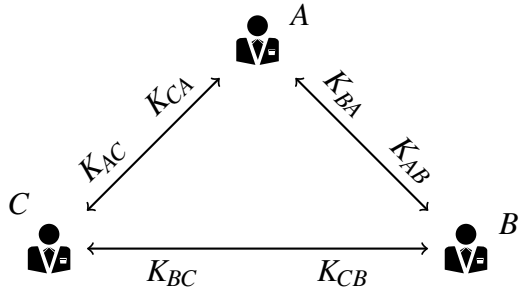


Figura 1.9: Distribuição de chaves em SDC-Net formando um dígrafo completo.

Similarmente, se A for um adversário, então A descobriria $\mathsf{H}(K_{BC}||j) \oplus \mathsf{H}(K_{CB}||j)$. Note que geraria mais aleatoriedade com duas funções de hash.

Se um dos usuários um adversário, ele pode apenas inverter sua mensagem, *i.e.*, calcular XOR 1 com sua mensagem, gerando no final das contas $1 \oplus 1 = 0$. Assim, os outros pensariam que um deles pagou quando nenhum deles pagou, e vice-versa. Somente quem pagou poderia perceber que há algo errado, mas revelando que há algo errado, revelaria sua identidade.

Poderíamos ter um problema diferente. Imagine que os três estão no elevador que somente permite levar 250Kg. Eles precisam saber a soma de seus pesos, mas não querem revelar seus respectivos pesos. Poderíamos querer saber a média da idade das pessoas que trabalham como criptógrafos, mas sem revelar idades individuais. Nestes casos, poderíamos usar uma operação de soma em vez de usarmos um XOR. Bastaria que os criptógrafos somassem as chaves enviadas e subtraíssem as chaves recebidas, ou vice-versa. [Borges de Oliveira 2017h] apresenta como SDC-Net pode ser usada em redes inteligentes.

Podemos generalizar como SDC-Nets funcionam. Inicialmente, cada usuário i envia uma chave para cada um dos outros. Para cada tempo j, cada usuário calcula

$$\mathfrak{M}_{i,j} = m_{i,j} + \sum_{o \in \mathscr{U} - \{i\}} \mathsf{H}\left(k_{i,o} || j\right) - \mathsf{H}\left(k_{o,i} || j\right),$$

onde $\mathscr{U}$ é o conjunto de usuários, $k_{i,o}$ é a chave enviada por i a o, e $k_{o,i}$ é a chave enviada por o a i.

Um usuário poderia ser o responsável por consolidar, descriptografar e revelar a consolidação c_j. No entanto, todos podem fazer isto executando uma SDC-Net. Além disto, o processo de consolidação já decripta, *i.e.*,

$$c_j = \sum_{i=1}^{I} \mathfrak{M}_{i,j}.$$

Independente da aplicação, SDC-Net permite que um adversário cause um rompimento, *i.e.*, um adversário pode inserir um valor falso invalidando o resultado do protocolo. Outro problema é o número de chaves que cresce quadraticamente, *i.e.*, para o caso da rede orientada temos $I(I-1)$ chaves, onde I é número de usuários, e para o caso da rede não-orientada temos

$$\frac{I(I-1)}{2}$$

chaves. Similarmente, o número total de operações que os usuários têm que calcular cresce quadraticamente com o número de usuários I. Para o caso da rede orientada temos $2I(I-1)$ operações e para o caso da rede não-orientada temos $I(I-1)$ operações, sem contarmos as operações referentes as mensagens.

Para diminuir a complexidade na distribuição de chaves e no processamento, os usuários podem enviar chaves apenas para quem eles confiam. Figura 1.10 esquematiza uma distribuição de chaves baseada em confiança. Note que o usuário B pode decifrar tudo que o usuário E encripta. Mas, ninguém pode decifrar as mensagens do usuário E sem a senha do usuário B. Os usuários têm que calcular menos funções de hash. O mais leve é E que calcula apenas uma função de hash, D calcula duas, C quatro, A cinco, e B seis. Basear a segurança em confiança não é uma boa escolha para um protocolo, mas é uma opção. Conectividade máxima evita conluio, mas tem um custo computacional.

Apesar da complexidade e da possibilidade de rompimento do protocolo, SDC-Net tem uma grande vantagem. Enquanto que PCHA permite que o vazamento de apenas uma mensagem encriptada $\mathfrak{M}_{i,j}$ para revelar a mensagem $m_{i,j}$ e o valor consolidado pode

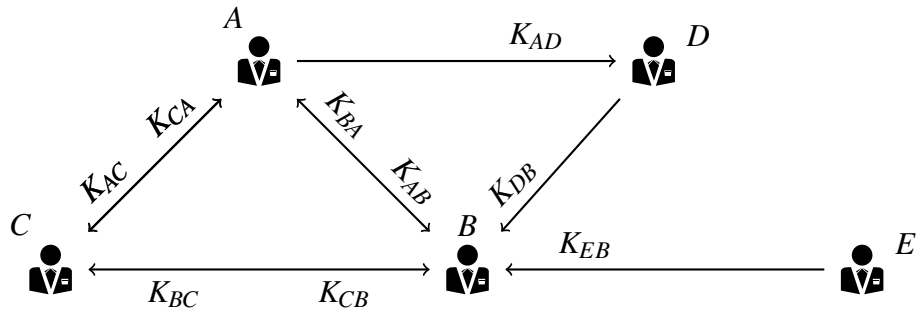


Figura 1.10: Distribuição de chaves em SDC-Net orientada.

não conter todas as contribuições, SDC-Net apresenta um cenário melhor. No caso do grafo completo, SDC-Net é resistente a conluio. Todos menos um $I-1$ devem conspirarem para descobrirem as mensagens de um usuário. Outra grande vantagem é que existe garantias que a informação consolidada somente será revelada com a contribuição de todos os usuários. SDC-Net apresenta o melhor cenário para a privacidade.

1.5. Técnicas Assimétricas e de Compromisso

Esta seção apresenta três técnicas: compromisso (*commitment*), encriptação homomórfica e DC-Nets assimétricas (ADC-Nets). O conhecimento matemático necessário para entender estas técnicas reside em estruturas algébricas que serão apresentadas antes das técnicas. Assim como funções de hash, técnicas de compromisso não são classificadas como simétricas ou assimétricas. Estão nesta seção apenas pela proximidade das ferramentas matemáticas usadas em técnicas de commitment e criptografia assimétrica. Em particular, o número de chaves de técnicas assimétricas cresce linearmente com o número de participantes. Entretanto, técnicas de commitment não tem chave criptográfica, mas precisam apenas armazenar um valor que se equipara a uma chave. Este valor é usado para provar—no sentido de garantir ou verificar—a mensagem. Diferente de técnicas criptográficas que se cifra uma mensagem e depois a decifra, commitment garante que o emissor de uma mensagem não vai alterar seu conteúdo, mas o receptor não o conhece até o emissor apresentá-lo. Logo, o receptor pode verificar uma soma sem conhecer as parcelas somadas com uma técnica de commitment homomórfico, por exemplo. Após a apresentação de técnicas de commitment, detalha-se Pedersen Commitments [Pedersen 1992] que garante uma segurança incondicional em relação a um commitment, ou seja, uma mensagem comprometida por um código criptográfico. Diferente da segurança incondicional em SDC-Nets, Pedersen Commitments é viável. Similarmente à apresentação de commitment, após a apresentação de técnicas de encriptação homomórfica, detalha-se o esquema de Paillier [Paillier 1999] que é uma técnica de encriptação homomórfica aditiva, e por isto, ela é usada em vários cenários. A seção termina detalhando SDC-Nets [Borges de Oliveira 2016]. Em particular, as equações de ADC-Nets podem ser derivadas das equações de técnicas de encriptação homomórfica aditiva. Logo, há semelhanças nas equações das técnicas, entretanto as propriedades das técnicas diferem em vários aspectos. O mesmo acontece com Pedersen Commitments que tem equação similar, mas não pode nem ser considerado um esquema que encripta e decripta.

1.5.1. Compromisso (Commitment)

Diferente de um sistema criptográfico que encripta e decripta mensagens $m_{i,j}$, técnicas de compromisso usam uma função de comprometimento Commit e uma função de abertura Open junto com um verificador randômico $\mathfrak{v}_{i,j}$, *i.e.*,

$$\mathfrak{N}_{i,j} = \mathsf{Commit}(m_{i,j}, \mathfrak{v}_{i,j})$$

e

$$\mathsf{Open}(\mathfrak{N}_{i,j}, m_{i,j}, \mathfrak{v}_{i,j}) = \top \veebar \bot,$$

onde ou a função de abertura retorna verdadeiro $\top$ ou retorna falso $\bot$. Desta forma, não é possível descriptografar a mensagem $m_{i,j}$, mas se pode verificar quando a mensagem $m_{i,j}$ é correta ou não.

Um exemplo de aplicação, seria um testamento $m_{i,j}$. Dado $\mathfrak{N}_{i,j}$ não se saberia qual é o testamento $m_{i,j}$, mas poderia verificar se ele é verdadeiro ou não. Sempre que pensarmos em auditoria, verificação, averiguação, podemos achar uma aplicação para técnicas de compromisso.

Em transações financeiras estamos interessados em verificar um total, *e.g.*, queremos verificar se a conta fecha no fim do mês. Neste caso, podemos proteger a privacidade dos usuários revelando apenas o valor final, *i.e.*, sem revelar as mensagens $m_{i,j}$ individuais. Abstratamente, cada usuário i no tempo j aplica a função

$$\mathfrak{N}_{i,j} = \mathsf{Commit}(m_{i,j}, \mathfrak{v}_{i,j})$$

e manda $\mathfrak{N}_{i,j}$ o resultado para um responsável pela verificação, digamos um auditor. Ao final de um período de tempo, *e.g.*, um mês, o auditor calcula

$$\mathfrak{U}_i = \prod_{j=1}^{J} \mathfrak{N}_{i,j}$$

enquanto cada usuário calcula

$$b_i = \sum_{j=1}^{J} m_{i,j}$$

e

$$\mathfrak{V}_i = \sum_{j=1}^{J} \mathfrak{v}_{i,j}$$

e envia $\mathfrak{U}_i$ e $\mathfrak{V}_i$ para o auditor, onde b_i é o valor da conta e $\mathfrak{U}_i$ é um verificador do usuário i gerado randomicamente. Para que o auditor verifique que b_i é o valor correto e que o usuário i mostre que é o correto, basta que eles calculem

$$\mathsf{Open}\left(\mathfrak{U}_i, b_i, \mathfrak{V}_i\right)$$

e verifiquem se o resultado é verdadeiro $\top$ ou se é falso $\bot$.

Se o auditor quisesse verificar a consolidação c_j para o número de usuários I, bastaria calcular

$$\mathfrak{T}_j = \prod_{i=1}^{I} \mathfrak{N}_{i,j}$$

e pedir o verificador no tempo j

$$\mathfrak{R}_j = \sum_{i=1}^{I} \mathfrak{v}_{i,j}.$$

Os usuários poderiam calcular o verificar com uma SDC-Net. Com as informações, o auditor verifica se

$$\mathsf{Open}\left(\mathfrak{T}_j, c_j, \mathfrak{R}_j\right)$$

retorna verdadeiro $\top$ ou se retorna falso $\bot$.

Em geral, um algoritmo de comprometimento não deve ser executado sozinho, mas junto com uma função de assinatura Sign que forneça uma assinatura digital $\mathfrak{S}_{i,j}$. Figura 1.11 esquematiza o modelo de comunicação entre usuários e auditor, que poderia ser um dos usuários. Cada mensagem comprometida

$$\mathfrak{N}_{i,j} = \mathsf{Commit}(m_{i,j}, \mathfrak{v}_{i,j})$$

deve ir concatenada a sua respectiva assinatura digital

$$\mathfrak{S}_{i,j} = \mathsf{Sign}\left(\mathfrak{N}_{i,j}\right).$$

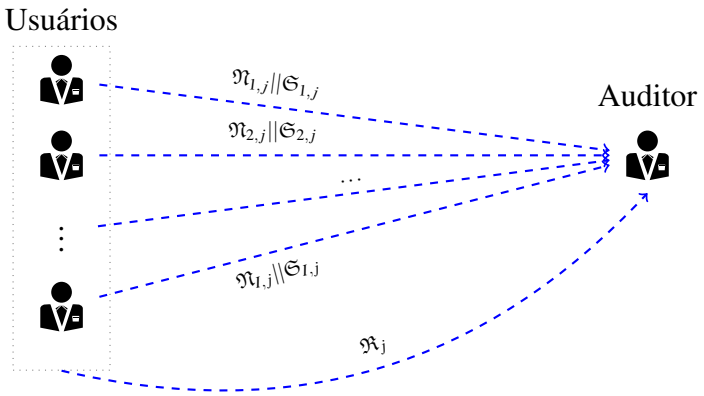


Figura 1.11: Modelo de comunicação para verificação no tempo j.

Entres as técnicas de comprometimento, existe uma que se sobressai porque pode prover segurança incondicional [Pedersen 1992]. Mas, antes precisamos saber que um grupo é um conjunto com uma operação que tem associatividade, identidade e inversa. Denotamos por $\mathbb{Z}_p$ o grupo multiplicativo dos números inteiros módulo um número primo p.

Antes dos usuários rodarem a função de comprometimento Commit, eles precisam estipular alguns valores em uma fase de inicialização, onde eles escolhem dois primos p e q tal que $q|(p-1)$ e sejam suficientemente grandes, e dado um gerador g de ordem q do subgrupo $G \subset \mathbb{Z}_p^*$, e deixam os valores públicos. Na sequência, cada usuário escolhe secretamente um $a_i \in \mathbb{Z}_q$ e calcula

$$h_i = g^{a_i} \pmod{p}$$

depois envia h_1 para o auditor. Note que dado h_i, g, e p, não se sabe da existência de um algoritmo clássico com tempo polinomial que determine a_i, *i.e.*, determinar a_i é um problema intratável para computadores clássicos. Porém, isso pode ser resolvido com o advento dos computadores quânticos [Shor 1997].

Para enviar a mensagem comprometida $\mathfrak{N}_{i,j}$, os usuários escolhem aleatoriamente outro valor secreto $\mathfrak{v}_{i,j} \in \mathbb{Z}_q$ e calculam

$$\mathfrak{N}_{i,j} = \mathsf{Commit}\left(m_{i,j}, \mathfrak{v}_{i,j}\right) = g^{m_{i,j}} h_i^{\mathfrak{v}_{i,j}} \mod p,$$

onde $m_{i,j} \in \mathbb{Z}_q$.

Para abrir, o auditor precisa da mensagem $m_{i,j}$ e do verificador randômico $\mathfrak{v}_{i,j}$. A função de abertura é definida por

$$\mathsf{Open}\left(\mathfrak{N}_{i,j}, m_{i,j}, \mathfrak{v}_{i,j}\right) = \left(\mathfrak{N}_{i,j} \stackrel{?}{=} g^{m_{i,j}} h_i^{\mathfrak{v}_{i,j}} \mod p\right).$$

Para garantir a privacidade a verificação deve acontecer em uma consolidação. Por exemplo, verificar o total das mensagens do usuário. Logo, o auditor calcula

$$\mathfrak{U}_i = \prod_{j=1}^{J} \mathfrak{N}_{i,j} = \prod_{j=1}^{J} g^{m_{i,j}} h_i^{\mathfrak{v}_{i,j}} \mod p$$

enquanto o usuário i calcula

$$b_i = \sum_{j=1}^{J} m_{i,j}$$

e

$$\mathfrak{V}_i = \sum_{j=1}^{J} \mathfrak{v}_{i,j}.$$

Na sequência, o usuário i envia $\mathfrak{U}_i$ e $\mathfrak{V}_i$ para o auditor. Para verificar se as somas estão corretas, basta que eles calculem

$$\mathsf{Open}\left(\mathfrak{U}_i, b_i, \mathfrak{V}_i\right) = \left(\mathfrak{U}_i \stackrel{?}{=} g^{b_i} h_i^{\mathfrak{V}_i} \mod p\right)$$

e verifiquem se o resultado é verdadeiro $\top$ ou se é falso $\bot$.

Para o auditor verificar a consolidação c_j no número de usuários I, temos um problema. O valor de h_i teria que ser igual para todos os usuários i. Mas, a_i deveria ser secreto. Então voltando a fase inicial, cada usuário poderia escolher seu a_i e usando uma SDC-Net, eles poderiam revelar o produto h s.t.

$$h = \prod_{i=1}^{I} g^{a_i}.$$

Então, eles geram um único h sem revelarem seus exponentes secretos a_i.

Desta forma, o auditor verifica a consolidação c_j no tempo j calculando

$$\mathfrak{T}_j = \prod_{i=1}^{I} \mathfrak{N}_{i,j} = \prod_{i=1}^{I} g^{m_{i,j}} h^{\mathfrak{v}_{i,j}} \mod p,$$

enquanto os usuários usam uma SDC-Net para calcular verificador no tempo j

$$\mathfrak{R}_j = \sum_{i=1}^{I} \mathfrak{v}_{i,j}.$$

Finalmente, eles verificam

$$\mathsf{Open}\left(\mathfrak{T}_j, c_j, \mathfrak{R}_j\right) = \left(\mathfrak{T}_j \stackrel{?}{=} g^{c_j} h^{\mathfrak{R}_j}\right)$$

retorna verdadeiro $\top$ ou se retorna falso $\bot$. A multi-exponenciação modular $g^{m_{i,j}} h^{\mathfrak{v}_{i,j}} \bmod p$ pode ser calculada rapidamente e paralelizada com ótimo balanceamento de cargas [Borges et al. 2017]. Para calcular a exponenciação ou multi-exponenciação modular, pode-se usar o Algoritmo 1 do apêndice A.

O processo de verificação por esquemas de compromissos pode ser feito de diversas maneiras. Um subconjunto de usuários poderia ser verificado. As consolidações no tempo poderiam ser semanais, mensais e anuais. Apenas, deve-se tomar cuidado que um excessivo número de verificações não leve a vazamentos de privacidade.

Um aspecto interessante de implementação é a geração de $\mathfrak{U}_i$, $\mathfrak{T}_j$, $\mathfrak{V}_i$, $\mathfrak{R}_j$, b_i e c_j pois o software não precisa armazenar todos os valores para gerá-los. É necessário armazenar apenas um valor para cada um formando um acumulador, *e.g.*, fazemos $c_j = 0$ e depois $c_j \leftarrow c_j + m_{i,j}$. Portanto, não se faz necessário armazenar todas as mensagens $m_{i,j}$. Após a verificação, também podemos eliminar c_j.

1.5.2. Encriptação Homomórfica

Existem dois tipos de encriptação homomórfica, a saber, parcialmente e completamente homomórficas. A primeira propicia uma das duas operações ou soma ou multiplicação sobre mensagem encriptada $\mathfrak{M}_{i,j}$. Enquanto que a segunda propicia ambas operações, *i.e.*, soma e multiplicação sobre mensagem encriptada $\mathfrak{M}_{i,j}$. Quando a técnica criptográfica apenas possibilita a operação de soma, dizemos que ela é uma primitiva de criptografia homomórfica aditiva (PCHA).

Atualmente, a maioria dos problemas de privacidade podem ser resolvidos apenas com PCHA. Logo, esta seção está focada em um sistema criptográfico desenvolvido por [Paillier 1999] que nos PCHA. Diferente de técnicas de compromisso que usam apenas um verificador randômico $\mathfrak{v}_{i,j}$, Paillier tem uma chave pública e outra privada. Logo, alguém seria responsável pela chave privada, digamos um contador.

Em uma fase inicial, o contador escolhe dois primos p e q grandes o suficiente de forma aleatória. Na sequência, o contador calcula $n = p \cdot q$ e $\lambda = \mathsf{mmc}(p-1, q-1)$, e escolhe aleatoriamente um número $g \in \mathbb{Z}^*_{n^2}$ s.t. n divide a ordem de g. Desta forma, o contador tem a chave privada (λ, μ) e distribui a chave pública (n, g) para os usuários, que podem encriptar com a função

$$\begin{aligned} &\mathsf{Enc} : \mathbb{Z}_n \times \mathbb{Z}^*_n \to \mathbb{Z}_{n^2} \\ &\mathsf{Enc}(m_{i,j}, r_{i,j}) \mapsto g^{m_{i,j}} \cdot {\mathfrak{v}_{i,j}}^n \mod n^2, \end{aligned} \tag{2}$$

onde $\mathfrak{v}_{i,j}$ é um valor randômico secreto. Normalmente, não escrevemos $\mathfrak{v}_{i,j}$. Logo, denotamos apenas $\mathfrak{M}_{i,j} = \mathsf{Enc}\left(m_{i,j}\right)$.

Para proteger a privacidade com PCHA, algum agregador semi-honesto deve entrar em ação e calcular

$$\mathfrak{C}_j = \sum_{i=1}^{I} \mathfrak{M}_{i,j}.$$

Semi-honesto significa que não forma um conluio e pode querer ler as mensagens $m_{i,j}$, mas o agregador não consegue decifrar as mensagens encriptadas $\mathfrak{M}_{i,j}$.

Para descriptografar o contador aplica a função

$$\begin{aligned} &\mathsf{Dec} : \mathbb{Z}_{n^2} \to \mathbb{Z}_n \\ &\mathsf{Dec}(\mathfrak{C}_j) \mapsto \mathrm{L}({\mathfrak{C}_j}^{\lambda} \mod n^2) \cdot d \mod n, \end{aligned} \tag{3}$$

onde $d = L(g^{\lambda} \mod n^2)^{-1}$.

Para se calcular a inversa multiplicativa em um grupo $\mathbb{Z}_n$, *i.e.*, de um inteiro módulo n, pode-se usar o Algoritmo 2 do apêndice A.

Note que a função que decripta Dec não usa o verificador randômico $\mathfrak{v}_{i,j}$. Por isto, não se faz necessário incluí-lo como parâmetro. Figura 1.12 esquematiza a comunicação de protocolos usando PCHA para proteger a privacidade. O agregador também poderia ser virtual, *i.e.*, a consolidação poderia ocorrer com os usuários enviando suas respectivas mensagens encriptadas $\mathfrak{M}_{i,j}$ uns para os outros de forma que um usuário tenha o produto de todas elas formando a consolidação encriptada $\mathfrak{C}_j$ e envie $\mathfrak{C}_j$ para o contador descriptografar.

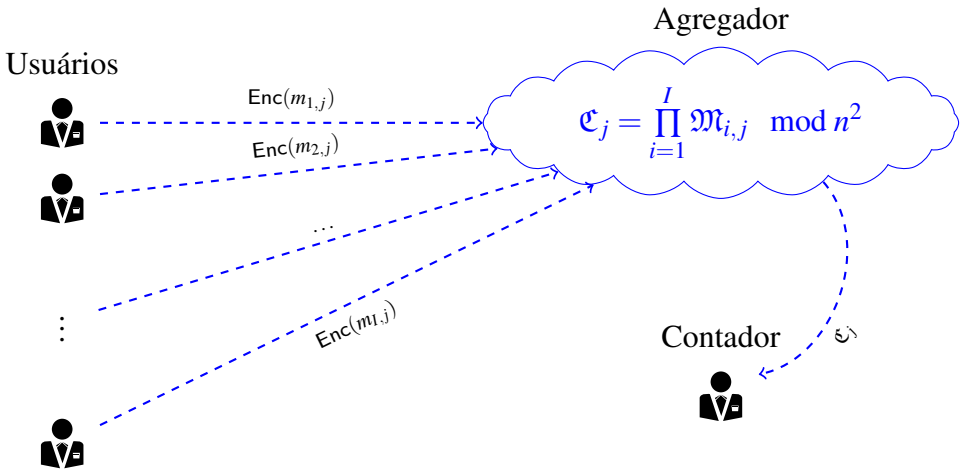


Figura 1.12: Modelo de comunicação para PCHA no tempo j.

Considerando a necessidade de um agregador, poderíamos construir uma SDC-Net em estrela com a propriedade de uma PCHA. Para medirmos e compararmos performance, SDC-Net em estrela apresenta um limite inferior e deve ser comparado com os novos protocolos [Borges de Oliveira 2017c]. Similarmente a SDC-Net, PCHA precisa de uma função de assinatura Sign. Em particular, pode-se usar alguma técnica de assinatura digital $\mathfrak{S}_{i,j}$ homomórfica, de forma que o agregador saiba quais usuários enviaram suas mensagens encriptadas $\mathfrak{M}_{i,j}$.

1.5.3. DC-Net Assimétrica

[Borges de Oliveira 2017g] apresenta o conceito de ADC-Net, que é similar ao de SDC-Net. Em particular, SDC-Net apresenta a possibilidade de verificação como técnicas de compromisso, e consequentemente, os usuários não podem romper o protocolo enviando um valor errado.

Em [Borges de Oliveira 2017g], um protocolo que protege a privacidade é definido como uma ADC-Net se satisfaz as seguintes propriedades:

1. o protocolo tem todas as propriedades de uma SDC-Net, excluindo segurança incondicional;
2. a segurança é baseada em uma função criptográfica *arapuca* do inglês *trapdoor*;
3. usuários podem usar chaves permanentes;
4. o tempo de processamento tem complexidade máxima polinomial;
5. não é necessário uma iteração sobre o número de usuários I, excluindo na consolidação;
6. usuários podem mandar o número mínimo de mensagens;
7. usuários podem usar uma função de assinatura para gerar uma assinatura digital $\mathfrak{S}_{i,j}$ de cada uma de suas mensagens $m_{i,j}$;
8. similar a uma técnica de comprometimento, usuários podem verificar suas mensagens $m_{i,j}$.

Baseado na propriedade 1, ADC-Net não precisa de um agregador. Dependendo do protocolo, ou todos podem descobrir o resultado da consolidação c_j, ou apenas um contador pode descobrir o resultado da consolidação c_j [Borges 2016]. Este capítulo usa basicamente a mesma ADC-Net apresentada em [Borges de Oliveira 2017g], com a diferença que todos os usuários enviam suas mensagens encriptadas para todos em vez de enviarem para um contador. Ambas os casos são equivalentes.

Durante o processo de inicialização do protocolo, os usuários escolhem um produto de primos n, por exemplo, como dado em [Boneh and Franklin 2001]. Cada usuário i escolhe uma chave privada k_i. Na sequência, eles determinam

$$s = \sum_{i=1}^{I} k_i,$$

de forma que todos saibam do valor de s sem revelar suas respectivas chaves privadas k_i, por exemplo, com uma SDC-Net.

Depois da configuração inicial, os usuários podem começar a cifrar suas mensagens com a função que encripta

$$\begin{aligned} &\mathsf{Enc} : \mathbb{Z}_n \to \mathbb{Z}_{n^2} \\ &\mathsf{Enc}_i(m_{i,j}) \mapsto (1+n)^{m_{i,j}} \cdot g^{h_j+k_i} \mod n^2, \end{aligned} \tag{4}$$

onde $h_j = \mathsf{H}(j)$ e H é uma função de hash s.t. se comporta como uma função de mão única e é resistente a colisões.

Os usuários i encriptam suas mensagens $m_{i,j}$ gerando mensagens encriptadas $\mathfrak{M}_{i,j}$ que são declaradas publicamente. Eles podem usar uma função de assinatura Sign para garantir a origem da mensagem. Caso algum usuário i não envie sua mensagem $m_{i,j}$ na janela de tempo j, todos sabem quem é o usuário i e podem requisitar que ele envie a respectiva mensagem encriptada $\mathfrak{M}_{i,j}$ com a assinatura digital $\mathfrak{S}_{i,j}$. No pior caso, os usuários podem inicializar novamente o protocolo excluindo quem não está enviando as mensagens encriptadas.

Para gerar a consolidação encriptada $\mathfrak{C}_j$ das mensagens encriptadas $\mathfrak{M}_{i,j}$, os usuários calculam

$$\mathfrak{C}_j = \prod_{i=1}^{I} \mathfrak{M}_{i,j} \mod n^2, \tag{5}$$

Note que o produtório que gera a consolidação encriptada $\mathfrak{C}_j$ pode ser calculado conforme as mensagens encriptadas $\mathfrak{M}_{i,j}$ vão chegando. Porém, o produtório somente resulta a consolidação encriptada $\mathfrak{C}_j$ após todas as mensagens encriptadas $\mathfrak{M}_{i,j}$ forem computadas. Consequentemente, a função que decripta Dec só pode ser executada após o final do produtório.

Para descriptografar a consolidação encriptada $\mathfrak{C}_j$, eles calculam

$$\begin{aligned} &\mathsf{Dec} : \mathbb{Z}_{n^2} \to \mathbb{Z}_n \\ &\mathsf{Dec}\left(\mathfrak{C}_j\right) \mapsto \frac{\left(\mathfrak{C}_j \cdot g^{-I \cdot h_j - s} \mod n^2\right) - 1}{n}, \end{aligned} \tag{6}$$

onde $s = \sum_{i=1}^{I} k_i$.

Pode-se mostrar que os protocolos usando Equações (4) a (6) geram uma ADC-Net, *i.e.*, satisfazem as oito propriedades descritas acima. É interessante notar que ADC-Nets são generalizações de PCHAs, *i.e.*, PCHAs são casos particulares de ADC-Nets. Pode-se criar ADC-Nets com equações que posam ser simplificadas de forma que resultem em PCHAs. Note também que Equação (4) poderia conter um fator randômico elevado a n, *i.e.*, $(\mathfrak{v}_{i,j})^n$, mas não se faz necessário.

O processo de construção de s poderia ser diferente. Em vez dos usuários gerarem s a partir de suas chaves privadas k_i, estas poderiam ser geradas para determinar um s fixo. Assim, ficaria fácil criar grupos de usuários confiáveis cuja soma das chaves dos membros de cada grupo de s. Para simplificar a função que decripta Dec na Equação (6), s poderia ser igual a n. O resultado das funções de hash H dos grupos poderiam ser diferentes, *i.e.*, existem várias formas de determinarmos grupos de usuários confiáveis em ADC-Net. Talvez a forma mais interessante seja quando a soma s é igual a zero para uma ADC-Net completa. Em particular, a soma s pode ser dada como na Figura 1.9 ou Figura 1.10.

Em técnicas de comprometimento, este capítulo descreve duas formas de verificação, a saber, consolidação comprometida do usuário i, *i.e.*, $\mathfrak{U}_i$ e consolidação comprometida do tempo j, *i.e.*, $\mathfrak{T}_j$. Com ADC-Nets, não precisamos verificar $\mathfrak{T}_j$, pois é possível descriptografar a consolidação encriptada $\mathfrak{C}_j$ e acessar a consolidação c_j no tempo j. Ao

descriptografar já temos a garantia que os valores estão comprometidos. Neste ponto, dependendo da aplicação, o protocolo que protege a privacidade pode verificar c_j com algum valor externo, *e.g.*, o número de votantes. Além disto, o usuário i pode fazer comprovações, *e.g.*, se ele votou nas últimas eleições. Para fazermos uma verificação sobre as mensagens do usuário i, calculamos

$$\mathfrak{U}_i = \prod_{j=1}^{J} \mathfrak{M}_{i,j}$$

e

$$\mathfrak{H} = \prod_{j=1}^{J} g^{h_j}$$

O usuário i calcula

$$\mathfrak{V}_i = \prod_{j=1}^{J} (1+n)^{m_{i,j}} \cdot g^{k_i} \mod n^2.$$

A função de abertura Open pode determina se os valores estão corretos, *i.e.*,

$$\mathsf{Open}\left(\mathfrak{U}_i, \mathfrak{V}_i, \mathfrak{H}\right) = \left(\mathfrak{U}_i \stackrel{?}{=} \mathfrak{V}_i \cdot \mathfrak{H}\right).$$

Se a função retornar verdadeiro, os valores estão corretos.

Diferente, poderíamos querer verificar o valor de consolidação por usuário b_i sem revelar as mensagens $m_{i,j}$ do usuário i. Desta forma, calculamos como anteriormente

$$\mathfrak{U}_i = \prod_{j=1}^{J} \mathfrak{M}_{i,j}.$$

Porém, o usuário i calcula

$$b_i = \sum_{j=1}^{J} m_{i,j}$$

e

$$\mathfrak{V}_i = \prod_{j=1}^{J} g^{h_j + k_i} \mod n^2.$$

A função de abertura Open pode tem que ser definida de forma diferente para determinar se os valores estão corretos, *i.e.*,

$$\mathsf{Open}\left(\mathfrak{U}_i, b_i, \right) = \left(\mathfrak{U}_i \stackrel{?}{=} (1+n)^{b_i} \cdot \mathfrak{V}_i\right).$$

A ADC-Net poderia ser construída para retornar o valor da consolidação por usuário b_i. No entanto, isto deve ser feito com muito cuidado para evitar o comprometimento do protocolo com o vazamento de informações privadas, *i.e.*, mensagens $m_{i,j}$.

Considerando verificações, seria mais aconselhável usar a função que encripta Enc dada por

$$\begin{aligned} \mathsf{Enc} &: \mathbb{Z}_n \to \mathbb{Z}_{n^2} \\ \mathsf{Enc}_i(m_{i,j}) &\mapsto (1+n)^{m_{i,j}} \cdot g^{h_j \cdot k_i} \mod n^2, \end{aligned}$$

em vez de Equação (4). A diferença está no produto dos expoentes, em vez da soma.

Logo, a respectiva função inversa, *i.e.*, a função que decripta Dec é dada por

$$\begin{aligned}&\mathsf{Dec} : \mathbb{Z}_{n^2} \to \mathbb{Z}_n \\ &\mathsf{Dec}\left(\mathfrak{C}_j\right) \mapsto \frac{\left(\mathfrak{C}_j \cdot g^{-h_j \cdot s} \mod n^2\right) - 1}{n},\end{aligned}$$

onde $s = \sum_{i=1}^{I} k_i$.

Similarmente, poderíamos construir diversas ADC-Nets, *e.g.*,

$$\begin{aligned}&\mathsf{Enc} : \mathbb{Z}_n \to \mathbb{Z}_{n^2} \\ &\mathsf{Enc}(m_{i,j}) \mapsto (1+n)^{m_{i,j}} \cdot h_j^{k_i} \mod n^2\end{aligned}$$

e

$$\begin{aligned}&\mathsf{Dec} : \mathbb{Z}_{n^2} \to \mathbb{Z}_n \\ &\mathsf{Dec}\left(\mathfrak{C}_j\right) \mapsto \frac{\left(\mathfrak{C}_j \cdot h_j^{-s} \mod n^2\right) - 1}{n}.\end{aligned}$$

Note que um número de usuários poderia se juntar para provar alguma propriedade do conjunto sem que as mensagens $m_{i,j}$ individuais deles sejam divulgadas. Similarmente, poderiam detectar um usuário querendo romper o protocolo que protege a privacidade. Por exemplo, suponha que um usuário i queira romper o protocolo, desta forma, ele poderia enviar um valor que preencha todos os bits de sua mensagem $m_{i,j}$ no tempo j. Se é feito uma consolidação c_j no tempo j e uma consolidação por usuário b_i por usuário i, então construir uma tabela com as consolidações e descobrir exatamente a mensagem $m_{i,j}$ que foi comprometida. A detecção do rompimento pode ser visualizada na Tabela 1.2. Mesmo que o usuário i enviasse várias mensagens comprometidas, eles seriam detectados com a Tabela 1.2.

O problema acontece quando não há duas consolidações ou quando não se deseja esperar a segunda consolidação para detectar a origem das mensagens que podem romper o protocolo. Outro problema acontece quando o usuário i envia uma mensagem $m_{i,j}$ pequena, mas que é inválida. Por exemplo, resultando que a soma dos votos seja maior que o número de votantes. Neste caso, pode-se separar os usuários em dois conjuntos $\mathscr{U}_1$ e $\mathscr{U}_2$. Na sequência, pode-se pedir aos usuários de cada conjunto para verificarem suas mensagens. Um dos dois grupos está com a soma errada resultando mais votantes que votos. O conjunto com erro pode ser dividido novamente, e para não comprometermos a privacidade, os usuários do conjunto não comprometido podem se juntar ao novo conjunto para ajudar na proteção da privacidade. Com este processo, podemos detectar o usuário i que tenta romper o protocolo em $\log_2(I)$ passos. Desta forma, temos

$$v = \sum_{i \in \mathscr{U}_1} m_{i,j}, \tag{7}$$

e

$$\mathfrak{V} = \prod_{i \in \mathscr{U}_1} g^{h_j + k_i} \mod n^2. \tag{8}$$

Tabela 1.2: Detectando a mensagem $m_{i,j}$ com um valor gigante.

	1	2	$\cdots$	j	$\cdots$	J	b_i
1	$m_{1,1}$	$m_{1,2}$	$\cdots$	$m_{1,j}$	$\cdots$	$m_{1,J}$	$\sum_{j=1}^{J} m_{1,j}$
2	$m_{2,1}$	$m_{2,2}$	$\cdots$	$m_{2,j}$	$\cdots$	$m_{2,J}$	$\sum_{j=1}^{J} m_{2,j}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i	$m_{i,1}$	$m_{i,2}$	$\cdots$	$m_{i,j}$	$\cdots$	$m_{i,J}$	$\sum_{j=1}^{J} m_{i,j}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
I	$m_{I,1}$	$m_{I,2}$	$\cdots$	$m_{I,j}$	$\cdots$	$m_{I,J}$	$\sum_{j=1}^{J} m_{I,j}$
c_j	$\sum_{i=1}^{I} m_{i,1}$	$\sum_{i=1}^{I} m_{i,2}$	$\cdots$	$\sum_{i=1}^{I} m_{i,j}$	$\cdots$	$\sum_{i=1}^{I} m_{i,J}$	$\sum_{j=1}^{j} c_j = \sum_{i=1}^{I} b_i$

O contador ou um interessando na verificação calcula

$$\mathfrak{P} = \prod_{i \in \mathscr{U}_1} \mathfrak{M}_{i,j} \tag{9}$$

Logo a função de abertura Open é definida por

$$\mathsf{Open}\left(\mathfrak{P}, \mathfrak{V}, v\right) = \left(\mathfrak{P} \stackrel{?}{=} (1+n)^v \cdot \mathfrak{V} \mod n^2\right). \tag{10}$$

O usuário que tenta romper o protocolo não tem escapatória, se ele está no conjunto $\mathscr{U}_1$, ou o valor de v não é o esperado, ou a função de abertura Open não vai abrir o comprometimento. Como ou o usuário está no conjunto $\mathscr{U}_2$ ou está no conjunto $\mathscr{U}_1$, consequentemente, se o conjunto $\mathscr{U}_1$ não tem problema, então ele está no conjunto $\mathscr{U}_2$.

Sem perda de generalidade, suponha que o problema está no $\mathscr{U}_1$, logo podemos separar os usuários do conjunto $\mathscr{U}_1$ em dois conjuntos $\mathscr{U}_{1_1}$ e $\mathscr{U}_{1_2}$, tal que

$$\mathscr{U}_{1_1} \cup \mathscr{U}_{1_2} = \mathscr{U}_1.$$

Para não comprometermos a privacidade podemos espalhar os usuários do $\mathscr{U}_2$ nos conjuntos $\mathscr{U}_{1_1}$ e $\mathscr{U}_{1_2}$, assim temos

$$\mathscr{U}_{1_1} \cup \mathscr{U}_{1_2} = \mathscr{U}_1 \cup \mathscr{U}_2.$$

Finalmente, calcula-se de forma semelhante as Equações (7) a (10), trocando apenas os conjuntos. Apesar de não se ter mais controle sobre o valor de v, tem-se o controle de quais usuários podem estar tentando romper o protocolo. O processo recursivo leva a detecção do usuário em $\log_2(I)$ passos.

1.6. Comparações

Esta seção apresenta uma comparação entre as melhores técnicas, enfatizando as semelhanças e diferenças, ou seja, lista-se qual técnica tem ou não uma propriedade. Por exemplo, que técnica garante verificação, ou garante que ninguém vai romper a privacidade, qual técnica é livre de terceiros confiáveis, etc. Especificamente, ADC-Nets podem garantir verificação de forma semelhante a commitment e ainda podem decriptar o resultado total das operações homomórficas, ou seja, computadas sobre os valores encriptados. Um outro ponto avaliado é a performance das técnicas apresentadas. Certamente, algoritmos que podem rodar com pouco processamento são fundamentais para equipamentos com restrições de hardware. No entanto, para computação ubíqua, usar menos processamento em bilhões de dispositivos significa economizar energia e produzir dispositivos com custo mais acessível. Nesta seção, pode-se ver que ADC-Nets superam as outras técnicas em performance além de proverem mais propriedades. Para isto, listamos a complexidade dos algoritmos na Tabela 1.3, considerando que o sistema criptográfico de Paillier tem o melhor desempenho das PCHA. Além do valor do número de usuários I e número de tempo J, a Tabela 1.3 contém n que é o produto de primos com no mínimo 1024 bits e k que é as chaves secretas de no mínimo 180 bits escolhidas pelos usuários. SDC-Net pode ser o mais rápido para um pequeno número de usuários I. Porém, vai ficando lento conforme o número de usuários I cresce. Técnicas de compromisso podem fazer verificações no número de usuários I e número de tempo J.

Tabela 1.3: Comparação da complexidade computacional.

Técnica	Enc	Consolidação	Dec
SDC-Net	$O(I)$	NA	$O(I)$
Compromisso	$O(\log(k))$	$O(J)$ ou $O(I)$	$O(k)$
PCHA	$O(\log(n))$	$O(I)$	$O(\log(n))$
ADC-Net	$O(\log(k))$	$O(I)$	$O(\log(k))$

Considerando os valores de n e k vemos que protocolos baseados em ADC-Net tendem a ser bem mais rápidos que protocolos baseados em PCHA. Acima de tudo, k cresce muito mais devagar que n quando se aumenta o nível de segurança. O número de bits das chaves para um dado nível de segurança é apresentado na Tabela 1.4. As diferenças são equivalentes as técnicas baseadas em curvas elípticas comparadas com o problema da fatoração de inteiros. Por isto, aparentemente não vale a pena construir uma ADC-Net baseada em curvas elípticas. Para uma comparação analítica da performance veja [Borges de Oliveira 2017a], e para uma comparação através de simulação, veja [Borges de Oliveira 2017i]. A Tabela 1.3 contém não aplicável (NA) na consolidação encriptada com SDC-Net porque o processo de consolidação acontece junto com a função que decripta Dec.

Podemos ver na Figura 1.13 que o número de bits de n tem um crescimento exponencial quando aumentamos o nível de segurança. Enquanto que o número de bits de k tem um crescimento linear com o aumento do nível de segurança. Uma vez que, a interpolação dos pontos da Tabela 1.4 nos fornece as curvas $y = 2x$ para os valores de k e $y = 601.53\exp(0.0127087x)$ para os valores de n. O tamanho da chave é diretamente pro-

Tabela 1.4: Comparação entre o crescimento de k e n com o nível de segurança.

Força Bruta nível de segurança	k	n
80	160	1 024
112	224	2 048
128	256	3 072
192	384	7 680
256	512	15 360

porcional ao custo computacional. Portanto, ADC-Net é cada vez mais rápida que PCHA quando o nível de segurança aumenta.

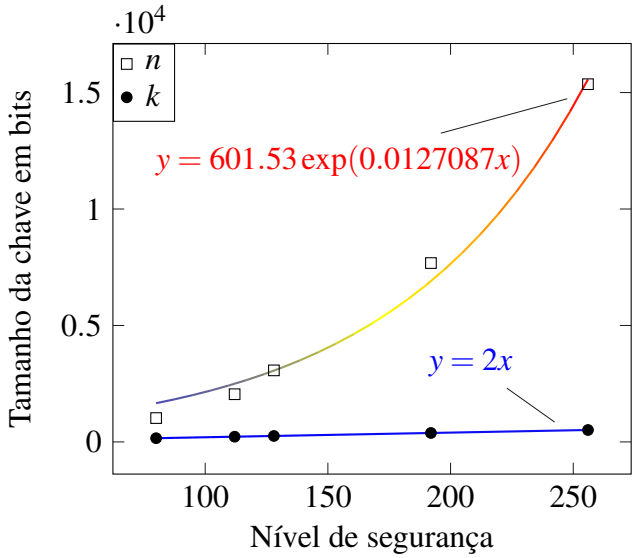


Figura 1.13: Curvas do crescimento de k e n em função do nível de segurança.

Algoritmos para SDC-Net tem diversas propriedades, ADC-Net tem mais ainda. Algoritmos de comprometimento podem ser completamente substituídos por algoritmos de ADC-Net. Neste caso, podemos comparar apenas as propriedades de SDC-Net, PCHA e ADC-Net. Uma propriedade importante é a capacidade de evitar conluio. Usando DC-Nets, podemos garantir que só se vaza as mensagens de um usuário quando todos estão contra ele, *i.e.*, $I-1$ usuários devem conspirar para revelarem mensagens de um usuário. No caso de PCHA, basta que um agregador envie mensagens encriptadas $\mathfrak{M}_{i,j}$ para um contador que possui a chave privada, e consequentemente, pode descriptografar as mensagens encriptadas $\mathfrak{M}_{i,j}$ que recebe. Logo basta o conluio de duas entidades. O valor de $I-1$ é o melhor possível em privacidade. DC-Nets tem o conceito de conjunto de usuários confiáveis, garantindo que mensagens encriptadas $\mathfrak{M}_{i,j}$ do mesmo conjunto de usuários somente serão descriptografadas juntas. Com os usuários confiáveis, o conluio de todos os outros não é suficiente para vazamento de informação, mais ainda, pode-se diminuir o tempo de processamento para SDC-Net.

Todas as três técnicas podem e devem usar uma função de assinatura Sign para ge-

rar uma assinatura digital $\mathfrak{S}_{i,j}$ de cada mensagem encriptada $\mathfrak{M}_{i,j}$. Porém, apenas usando DC-Nets, o destinatário pode verificar quem enviou ou não e se a assinatura digital $\mathfrak{S}_{i,j}$ está correta, pois ADC-Net necessita de um agregador. Consequentemente, os usuários podem descriptografar a consolidação c_j com DC-Nets, mas não podem com PCHA.

Pode-se construir protocolos que protejam a privacidade que enviam o mínimo número de mensagens, mas o tempo de processamento não é constante em todas as técnicas. Conforme o número de usuários I cresce, protocolos baseados em SDC-Net ficam lentos, logo eles não são escaláveis com o número de usuários. Quanto ao processo da consolidação encriptada $\mathfrak{C}_j$ também depender do número de usuários I, esta faz operações bem mais simples e cresce linearmente com o número de usuários I, já função que encripta Enc cresce quadraticamente no total, apesar de linearmente para cada usuário.

Todas as técnicas permitem o uso de chaves permanentes, *i.e.*, chaves que podem ser geradas em um processo inicial e usadas para gerar várias mensagens encriptadas $\mathfrak{M}_{i,j}$. Porém, o número de chaves varia de acordo com a técnica. Cada usuário em uma SDC-Net completa precisa armazenar $2(I-1)$ chaves, mas o total de chaves no protocolo cresce quadraticamente $O(I^2)$ com o número de usuários I. Já usando PCHA, cada i tem uma chave criptográfica e existem apenas duas chaves criptográficas no protocolo, a saber, a pública que encripta e a privada que decripta. Com ADC-Net a situação é diferente. Cada usuário tem uma chave privada única, mas o número total de chaves no protocolo cresce linearmente com o número de usuários I. Apesar de protocolos baseados em ADC-Net terem mais chaves que protocolos baseados em PCHA, eles têm o menor número de chaves para garantir a privacidade e evitar o conluio. Usando ADC-Net, cada usuário tem sua chave privada e a chave inversa para descriptografar pode ser pública ou privada.

Pode-se projetar protocolos que rodem em tempo polinomial com as três técnicas. Porém, somente ADC-Net possibilita verificação de mensagens encriptadas $\mathfrak{M}_{i,j}$ sem violar a privacidade dos usuários. Consequentemente, nenhum usuário i pode tentar romper o protocolo enviando mensagens $m_{i,j}$ erradas que ele será detectado. Em teoria, técnicas de PCHA poderiam possibilitar usuários a verificar se enviaram a mensagem encriptada $\mathfrak{M}_{i,j}$ correta, mas quem possui a chave privada não poderia verificar, pois não poderia receber as mensagens $m_{i,j}$ sem possibilidade de acessar as mensagens $m_{i,j}$.

Excluindo a complexidade computacional já apresentada na Tabela 1.3, apresenta-se na Tabela 1.5 um resumo das propriedades discutidas nesta seção.

1.7. Considerações Finais

Esta seção enfatiza os pontos principais das seções anteriores junto com considerações sobre as técnicas apresentadas neste capítulo. Acima de tudo, esta seção apresenta as limitações encontradas nas técnicas com suas primitivas criptográficas, os desafios encontrados nos cenários e uma perspectiva de futuros trabalhos voltando a discutir os cenários práticos de aplicação de proteção à privacidade.

Seção 1.1 tem apresentado uma visão geral deste capítulo começando a introduzir a necessidade de preservarmos a privacidade, enquanto que a Seção 1.2 tem apresentado diversos cenários onde se faz necessário a preservação da privacidade. Seção 1.3 tem

Tabela 1.5: Comparação das propriedades.

Propriedades	**SDC-Net**	**PCHA**	**ADC-Net**
Evita conluio	✓		✓
Conjunto de usuários confiáveis	✓		✓
Mensagens direto para o destinatário	✓		✓
Usuários podem descriptografar	✓		✓
Número mínimo de mensagens	✓	✓	✓
Escalável		✓	✓
Chaves permanentes	✓	✓	✓
Baseado em *trapdoors*	✓	✓	✓
Chaves armazenadas por usuário	$2(I-1)$	1	1
Total de chaves	$O(I^2)$	2	$O(I)$
Tempo polinomial	✓	✓	✓
Possibilidade de verificação			✓
Não se pode romper o protocolo			✓

apresentado como podemos avaliar as técnicas usadas para proteger a privacidade. Em particular, uma análise meramente das técnicas não apresenta qual a probabilidade de cada mensagem. Considerando que as melhores técnicas devem deixar as mensagens equiprováveis, devemos analisar a probabilidade de se inferir as mensagens com os dados que podem ser encontrados no problema de cada cenário. Seção 1.4 tem apresentado as técnicas simétricas para proteger a privacidade, enquanto que a Seção 1.5 tem apresentado as técnicas assimétricas. A comparação entre os pontos mais importantes das técnicas é apresentada na Seção 1.6.

SDC-Net é a técnica simétrica que apresenta mais propriedades, logo a mais interessante. Entre elas, ADC-Net pode garantir segurança incondicional, mas os usuários só podem usar suas chaves uma vez. Podemos usar a chaves várias vezes usando uma função de hash, que apesar de ser rápida, o tempo de processamento para encriptar usando SDC-Net cresce quadraticamente com o número de usuários.

As técnicas assimétricas têm um tempo de processamento independentemente do número de usuários. Além disto, cada usuário tem apenas uma chave. Usando PCHA, todos têm a mesma chave pública, mas somente uma entidade tem a chave privada que pode decifrar todas as mensagens. Por esta razão, tal entidade não pode ter acesso ao processo de consolidação encriptada. Ainda mais, ninguém que tenha acesso ao processo pode criar um conluio com tal entidade. Portanto, os usuários não têm garantia de privacidade, *i.e.*, que suas mensagens encriptadas não serão decriptadas. Diferentemente, os usuários têm garantia que suas mensagens encriptadas não serão decifradas individualmente quando eles usam DC-Nets. Além desta propriedade, ADC-Net tem todas as propriedades de SDC-Net. Em adição, ADC-Net também possibilita verificações de forma semelhante que em técnicas de comprometimento, e consequentemente, os usuários não podem romper o protocolo enviando uma mensagem inválida. ADC-Net força a garantia da privacidade e possibilita auditorias mantendo a privacidade.

A fase de inicialização com PCHA é mais simples do que com ADC-Net. Com a

primeira técnica, todos os usuários recebem a mesma chave pública de quem tem a chave privada. Porém, isto gera insegurança. Com a segunda técnica, todos os usuários devem escolher suas respectivas chaves privadas e juntos gerarem a chave inversa que pode ser pública ou privada, mas que decripta apenas a consolidação encriptada, *i.e.*, mensagens individuais não podem ser decriptadas. A fase de inicialização de protocolos com ADC-Net poderia ser tão simples como com PCHA se for introduzido uma autoridade confiável para distribuir chaves. Porém, tal autoridade seria um ponto crítico de falha. Assim, como a entidade que detém a chave privada de uma técnica de PCHA é um ponto crítico de falha.

A inserção e remoção de usuários nos protocolos é bem mais simples com PCHA do que com DC-Nets. Basta enviar a chave pública e revogar o reconhecimento da assinatura digital para técnicas de PCHA. Para o caso de DC-Nets, os protocolos devem ser reinicializados. Aqui existe uma relação inversa entre praticidade e privacidade.

Diversos cenários onde se deve preservar a privacidade precisam da operação de soma, *e.g.*, votação eletrônica, sistemas de reputação, redes de sensores, cibermedicina, processamento de imagens, dinheiro eletrônico, computação em múltiplas partes, privacidade no mundo acadêmico, redes inteligentes *etc.* Em cibermedicina, encontramos operações com textos, por exemplo, no prontuário eletrônico. DC-Nets também podem trabalhar com textos criando uma rede de anonimato. No entanto, buscas em textos encriptados e encriptação com múltiplas chaves têm maiores aplicações em textos que somas. Acima de tudo, devemos considerar que se protegermos uma operação de soma em qualquer cenário, então estaremos protegendo todas as operações subsequentes. Este é o processo de proteção de alguns protocolos de proteção da privacidade em processamento de imagens.

A definição do problema é um dos desafios encontrados nos cenários para proteção da privacidade. Por exemplo, se tentarmos proteger um dado médico, teremos que proteger todos os lugares onde ele aparece, mas nem sempre está claro onde este dado aparece. Mais ainda, a relação entre outros dados pode levar a dedução do dado protegido. Todos os cenários parecem ter uma relação inversa entre privacidade e disponibilidade da informação. No entanto, esta relação inversa não necessariamente existe. Muitos cenários podem ter a privacidade protegida com a informação disponível no momento certo. Quando se atinge a proteção almejada, temos que começar os processos de otimização para reduzir custos de processamento.

Protocolos baseados em ADC-Net podem requisitar menor espaço para armazenamento das chaves e serem cada vez mais rápidos que PCHA quando o nível de segurança aumenta. Aparentemente, não existe vantagem em desenvolver ADC-Net baseadas em outras primitivas, *e.g.*, curvas elípticas. O tempo de processamento ótimo para encriptar mensagens para gerar uma consolidação é alcançado com uma SDC-Net estrela, *i.e.*, $O(1)$. Portanto, quaisquer novas técnicas deveriam ter a performance comparada com uma SDC-Net estrela e o estado da arte de ADC-Net.

O uso de algoritmos criptográficos para proteger a privacidade é relativamente novo. Em particular, SDC-Net foi criada quase trinta anos antes da recém-criada ADC-Net. Todos os cenários onde se deseja proteger a privacidade devem ser revisitados com novas ferramentas, em especial com ADC-Net.

Como tecnologias da informação e comunicação estão continuamente conectando pessoas, dados e dispositivos, temos cada dia mais um maior volume de dados para tratar e informações privadas para proteger. O vazamento de tais informações tem impacto direto na vida de cada cidadão. Muitos problemas estão em aberto ou precisam ser otimizados. Faz-se necessário muita pesquisa na área de privacidade.

A. Algoritmos

Algorithm 1: Multi-exponenciação modular

Input: Inteiros b_i, e_i, m, n s.t. $e_i = \sum_{j=1}^{l_i} 2^{j-1} e_{i_j}$, onde $l_i = \lceil \log_2 e_i \rceil$ e $e_{i_j} \in \{0,1\}$.

Output: $\prod_{i=1}^{n} b_i^{e_i} \mod m$.

```
1 L ← ⌈max(log2 e1, ..., log2 en)⌉
2 a ← 1
3 for j = L to 1 by −1 do
4     a ← a^2 mod m
5     for i = 1 to n do
6         if e_{i_j} = 1 then
7             a ← a · b_i mod m
8 return a
```

Algorithm 2: Inversa multiplicativa do elemento a de um grupo $\mathbb{Z}_n$ usando o Algoritmo Euclidiano Estendido

Input: Inteiros a e n
Output: $a^{-1} \mod n$

$a \leftarrow a \mod n$
$t \leftarrow 0$
$t' \leftarrow 1$
$r \leftarrow n$
$r' \leftarrow a$
while $r' \neq 0$ **do**
  $q \leftarrow \lfloor r/r' \rfloor$
  $t \leftarrow t'$
  $t' \leftarrow t - q \cdot t'$
  $r \leftarrow r'$
  $r' \leftarrow r - q \cdot r'$
if $r > 1$ **then**
  return "a não tem inversa"
if $t < 0$ **then**
  $t \leftarrow t + n$
return t

B. Lista de Acrônimos

ADC-Net DC-Net assimétrica. 21, 27–30, 32–36

DC-Net *Dining Cryptographers Network*. 16, 33–36

NA não aplicável. 32

PCHA primitiva de criptografia homomórfica aditiva. 12, 13, 20, 25, 26, 28, 32–36
PMU *phasor measurement unit*. 12

SDC-Net DC-Net simétrica. 16, 18–21, 23–27, 32–36

C. Lista de Abreviações

e.g. "por exemplo" de *exempli gratia* em Latim. 3–5, 7, 16–19, 22, 25, 29, 30, 36, 40
etc. "e outros" ou "e assim por diante" de *et cetera* em Latim. 4, 6, 7, 13, 36, 40

i.e. "isto é" ou "ou seja" de *id est* em Latim. 2, 5, 6, 8, 10, 12–15, 17–20, 22, 24–26, 28–30, 33–36, 39, 40
iff condição necessária e suficiente de "if and only if" em Inglês. 39

s.t. "tal que" de *such that* em Inglês. 24, 25, 28, 37, 39

v. "contra" ou "em contraste com" de *versus* em Latim. 4

D. Lista de Símbolos

assinatura digital ($\mathfrak{S}_{i,j}$) assinatura digital do usuário i no tempo j. 23, 26–28, 34, 36, 39
atribuição ($\leftarrow$) $a \leftarrow a+1$ significa a atribuição de $a+1$ para a. 25, 37–39

caso ($\stackrel{?}{=}$) os valores são corretos quando a equação é satisfeita. 24, 25, 29, 31
consolidação (c_j) consolidação das mensagens agregadas no tempo j, *i.e.*, $c_j = \mathsf{Dec}\left(\mathfrak{C}_j\right)$. 12, 13, 20, 22–32, 34, 36, 39
consolidação comprometida do usuário i ($\mathfrak{U}_i$) consolidação das mensagens comprometidas no número de tempo J, *i.e.*, $\mathfrak{U}_i = \prod_{j=1}^{J} \mathfrak{N}_{i,j}$. 22, 24, 25, 28, 29, 39
consolidação comprometida do tempo j ($\mathfrak{T}_j$) consolidação das mensagens comprometidas do número de usuários I, *i.e.*, $\mathfrak{T}_j = \prod_{i=1}^{I} \mathfrak{N}_{i,j}$. 22–25, 28, 39
consolidação encriptada ($\mathfrak{C}_j$) consolidação encriptada das mensagens no tempo j, s.t. $\mathfrak{C}_j = \prod_{j=1}^{J} \mathfrak{M}_{i,j}$. 12, 13, 26, 28, 30, 32, 34–36, 39
consolidação por usuário (b_i) consolidação das mensagens comprometidas pelo do usuário i, *i.e.*, $b_i = \sum_{j=1}^{J} m_{i,j}$. 22, 24, 25, 29–31

função de abertura (Open) a função que abre mensagem comprometida $\mathfrak{N}_{i,j}$ e retorna verdade **iff** os valores estão corretos. 22–25, 29, 31
função de assinatura (Sign) função que retorna uma assinatura digital $\mathfrak{S}_{i,j}$. 23, 26–28, 33
função de comprometimento (Commit) um esquema de comprometimento definido de acordo com o protocolo. 22–24, 39
função de hash (H) uma função de hash s.t. se comporta como uma função de mão única e é resistente a colisões. 19, 20, 28, 35
função que decripta (Dec) uma função que decripta definida de acordo com o protocolo. 11–13, 26, 28, 30, 32, 39
função que encripta (Enc) uma função que encripta definida de acordo como protocolo. 11–13, 25–27, 29, 30, 32, 34, 39

inteiros ($\mathbb{Z}$) o conjunto dos números inteiros. 23–30, 38

mensagem ($m_{i,j}$) abstração de um dados ou informação do usuário i no tempo j. 3, 10–13, 15–31, 33–36, 39, 40
mensagem comprometida ($\mathfrak{N}_{i,j}$) mensagem comprometida de $m_{i,j}$, *i.e.*, $\mathfrak{N}_{i,j} = \mathsf{Commit}\left(m_{i,j}, \mathfrak{v}_{i,j}\right)$. 22–24, 39
mensagem encriptada ($\mathfrak{M}_{i,j}$) mensagem cifrada do usuário i no tempo j, *i.e.*, $\mathfrak{M}_{i,j} = \mathsf{Enc}\left(m_{i,j}\right)$. 11–13, 16, 17, 20, 25–29, 31, 33–35, 39

mínimo múltiplo comum (mmc) função que retorna o mínimo múltiplo comum. 25

número de tempo (J) número total de tempo j. 12, 22, 24, 29, 31, 32, 39, 40
número de usuários (I) número total de usuários i. 12, 13, 20–28, 30–35, 39, 40

tempo (j) identificação do tempo. 11, 12, 17, 19, 20, 22–32, 39, 40

usuário (i) abstração de um sistema computacional e seu respectivo usuário com identificação i. 3–5, 10–12, 16–36, 39, 40

verificador do usuário i ($\mathfrak{V}_i$) consolidação das mensagens comprometidas no número de tempo J, *i.e.*, $\mathfrak{V}_i = \prod_{j=1}^{J} \mathfrak{v}_{i,j}$. 22, 24, 25, 29, 40
verificador no tempo j ($\mathfrak{R}_j$) consolidação das mensagens comprometidas do número de usuários I, *i.e.*, $\mathfrak{R}_j = \prod_{i=1}^{I} \mathfrak{v}_{i,j}$. 23, 25, 40
verificador randômico ($\mathfrak{v}_{i,j}$) verificador aleatório da mensagem do usuário i no tempo j. 22–26, 28, 39, 40

E. Glossário

adversário uma abstração de entidades adversárias, atacantes, criminoso, *etc.* que tenta descobrir a mensagem $m_{i,j}$. 4–7, 11–13, 15–20

destinatário uma abstração da entidade que recebe a mensagem $m_{i,j}$, *e.g.*, receptor, destino, sumidouro, consumidor, *etc.* em geral um usuário. 11–13, 17, 34, 35

função de mão única se existe, é uma função que pode ser computada em tempo polinomial, mas sua inversa não pode. 28, 39

remetente uma abstração da entidade que envia a mensagem $m_{i,j}$, *e.g.*, emissor, origem, fonte, produtor, *etc.* em geral um usuário. 12, 13, 17

Referências

[Al Ameen et al. 2012] Al Ameen, M., Liu, J., and Kwak, K. (2012). Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of Medical Systems*, 36(1):93–101.

[Bellare et al. 2007] Bellare, M., Boldyreva, A., and O'Neill, A. (2007). *Deterministic and Efficiently Searchable Encryption*, pages 535–552. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Boneh and Franklin 2001] Boneh, D. and Franklin, M. (2001). Efficient generation of shared rsa keys. *J. ACM*, 48(4):702–722.

[Borges 2016] Borges, F. (2016). *Privacy-Preserving Data Aggregation in Smart Metering Systems*. Energy Engineering Series. Institution of Engineering & Technology.

[Borges et al. 2017] Borges, F., Lara, P., and Portugal, R. (2017). Parallel algorithms for modular multi-exponentiation. *Applied Mathematics and Computation*, 292:406 – 416.

[Borges et al. 2012] Borges, F., Martucci, L. A., and Mühlhäuser, M. (2012). Analysis of privacy-enhancing protocols based on anonymity networks. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pages 378–383.

[Borges de Oliveira 2016] Borges de Oliveira, F. (2016). *On Privacy-Preserving Protocols for Smart Metering Systems: Security and Privacy in Smart Grids*. Springer International Publishing.

[Borges de Oliveira 2017a] Borges de Oliveira, F. (2017a). *Analytical Comparison*, pages 101–110. Springer International Publishing, Cham.

[Borges de Oliveira 2017b] Borges de Oliveira, F. (2017b). *Background and Models*, pages 13–23. Springer International Publishing, Cham.

[Borges de Oliveira 2017c] Borges de Oliveira, F. (2017c). *Concluding Remarks*, pages 127–129. Springer International Publishing, Cham.

[Borges de Oliveira 2017d] Borges de Oliveira, F. (2017d). *Introduction*, pages 3–12. Springer International Publishing, Cham.

[Borges de Oliveira 2017e] Borges de Oliveira, F. (2017e). *Quantifying the Aggregation Size*, pages 49–60. Springer International Publishing, Cham.

[Borges de Oliveira 2017f] Borges de Oliveira, F. (2017f). *Reasons to Measure Frequently and Their Requirements*, pages 39–47. Springer International Publishing, Cham.

[Borges de Oliveira 2017g] Borges de Oliveira, F. (2017g). *Selected Privacy-Preserving Protocols*, pages 61–100. Springer International Publishing, Cham.

[Borges de Oliveira 2017h] Borges de Oliveira, F. (2017h). *A Selective Review*, pages 25–36. Springer International Publishing, Cham.

[Borges de Oliveira 2017i] Borges de Oliveira, F. (2017i). *Simulation and Validation*, pages 111–126. Springer International Publishing, Cham.

[Camenisch et al. 2007] Camenisch, J., Lysyanskaya, A., and Meyerovich, M. (2007). Endorsed e-cash. In *Security and Privacy, 2007. SP '07. IEEE Symposium on*, pages 101–115.

[Chan and Perrig 2003] Chan, H. and Perrig, A. (2003). Security and privacy in sensor networks. *Computer*, 36(10):103–105.

[Chaum 1988] Chaum, D. (1988). The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptol.*, 1(1):65–75.

[Chaum 1981] Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90.

[Cramer et al. 2001] Cramer, R., Damgård, I., and Nielsen, J. B. (2001). Multiparty computation from threshold homomorphic encryption. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '01, pages 280–299, London, UK, UK. Springer-Verlag.

[Cramer et al. 1997] Cramer, R., Gennaro, R., and Schoenmakers, B. (1997). A secure and optimally efficient multi-authority election scheme. In *Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'97, pages 103–118, Berlin, Heidelberg. Springer-Verlag.

[De Montjoye et al. 2013] De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3.

[Díaz et al. 2003] Díaz, C., Seys, S., Claessens, J., and Preneel, B. (2003). Towards measuring anonymity. In *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 54–68, Berlin, Heidelberg. Springer-Verlag.

[Dwork 2008] Dwork, C. (2008). Differential privacy: A survey of results. In Agrawal, M., Du, D., Duan, Z., and Li, A., editors, *Theory and Applications of Models of Computation*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer Berlin Heidelberg.

[El Gamal 1985] El Gamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 10–18, New York, NY, USA. Springer-Verlag New York, Inc.

[Farhi et al. 2012] Farhi, E., Gosset, D., Hassidim, A., Lutomirski, A., and Shor, P. (2012). Quantum money from knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 276–289, New York, NY, USA. ACM.

[Gritzalis 2002] Gritzalis, D. A. (2002). Principles and requirements for a secure e-voting system. *Computers & Security*, 21(6):539 – 556.

[Jøsang et al. 2007] Jøsang, A., Ismail, R., and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618 – 644. Emerging Issues in Collaborative Commerce.

[Kerschbaum 2009] Kerschbaum, F. (2009). A verifiable, centralized, coercion-free reputation system. In *Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society*, WPES '09, pages 61–70, New York, NY, USA. ACM.

[Li et al. 2007] Li, N., Li, T., and Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115.

[Li and Cao 2013] Li, Q. and Cao, G. (2013). Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error. In De Cristofaro, E. and Wright,

M., editors, *Privacy Enhancing Technologies*, volume 7981 of *Lecture Notes in Computer Science*, pages 60–81. Springer Berlin Heidelberg.

[Naor and Shamir 1995] Naor, M. and Shamir, A. (1995). *Visual cryptography*, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Paillier 1999] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer.

[Pedersen 1992] Pedersen, T. P. (1992). Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 129–140, London, UK, UK. Springer-Verlag.

[Peter et al. 2013] Peter, A., Tews, E., and Katzenbeisser, S. (2013). Efficiently outsourcing multiparty computation under multiple keys. *IEEE Transactions on Information Forensics and Security*, 8(12):2046–2058.

[Peter et al. 2010] Peter, S., Westhoff, D., and Castelluccia, C. (2010). A survey on the encryption of convergecast traffic with in-network processing. *Dependable and Secure Computing, IEEE Transactions on*, 7(1):20–34.

[Reid and Harrigan 2013] Reid, F. and Harrigan, M. (2013). *An Analysis of Anonymity in the Bitcoin System*, pages 197–223. Springer New York, New York, NY.

[Santini 2005] Santini, S. (2005). We are sorry to inform you ... *Computer*, 38(12):128–127.

[Shor 1997] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509.

[Vaccaro et al. 2007] Vaccaro, J. A., Spring, J., and Chefles, A. (2007). Quantum protocols for anonymous voting and surveying. *Phys. Rev. A*, 75:012333.

[Wang and Yu 2005] Wang, X. and Yu, H. (2005). How to break md5 and other hash functions. In *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'05, pages 19–35, Berlin, Heidelberg. Springer-Verlag.

[Zheng and Huang 2013] Zheng, P. and Huang, J. (2013). An efficient image homomorphic encryption scheme with small ciphertext expansion. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 803–812, New York, NY, USA. ACM.

Capítulo 2

Crimes Cibernéticos e Computação Forense

Wilson Leite da Silva Filho

Abstract

Cibercrimes have caused a deep impact in the society. They are among the three types of crimes that have caused the major financial loss in the world, staying only behind of traffic of drugs and falsification. From the investigation and production of proves needs against this type of crime, rises the computer forensics area. This area is responsible for collect, preserve, process and present its results to the legal authorities. Its a computer science area in continues development, that demands an ongoing research of theirs experts, once each new digital technology is also an opportunity to commit crimes.

Resumo

Crimes cibernéticos têm causado um impacto cada vez maior na sociedade. Está entre os três tipos de crimes que causam maior prejuízo financeiro no mundo, ficando atrás apenas do tráfico de drogas e a falsificação. Da necessidade de investigação e produção de provas para o combate a este ilícito, surge a área de computação forense. Esta área provê técnicas para a coleta, preservação, processamento e apresentação de evidências para autoridades legais. É uma área da Ciência da Computação em constante desenvolvimento e que demanda pesquisa e atualização contínua dos especialistas, na qual cada nova tecnologia emergente traz consigo o potencial de também ser explorada para fins ilícitos.

2.1. Introdução

O objetivo deste material didático é apresentar aos alunos a área de Computação Forense, com foco na área criminal, englobando alguns dos principais crimes

cibernéticos e as técnicas e ferramentas usadas na área. O texto é embasado na literatura e nos conhecimentos empíricos das perícias criminais oficiais de informática realizadas no Estado de Santa Catarina pelo Instituto de Criminalística do IGP/SC.

O texto está estruturado em oito seções. A seção 2.1 é esta introdução. Na seção 2.2, Crimes Cibernéticos, é apresentada uma definição de crimes que envolvem a área de informática e são mostrados exemplos de crimes cibernéticos de repercussão e cifras correspondentes ao prejuízo causado por este tipo de ato. O objetivo principal é sensibilizar o leitor acerca da importância e dimensão que esse tipo de delito possui nos dias atuais.

Na seção 2.3, Princípios da Computação Forense, são abordadas as principais etapas do processo de perícia em artefatos digitais, bem como, são enfatizadas as precauções necessárias para a correta aquisição e manipulação da evidência digital, mantendo-a válida durante todo o processo legal.

Aspectos Jurídicos em Computação Forense é o assunto da seção 2.4. São apresentadas, de forma sucinta, algumas das leis que estão diretamente relacionadas à área de computação forense, com foco na área criminal.

Na seção 2.5, Laboratório de Computação Forense: Preservação e Análise da Prova Digital, são apresentados os dispositivos de *hardware* e *software* usados em um laboratório de forense computacional. São mostrados *softwares* comerciais e *softwares* livres para diversas atividades dessa área.

Em Princípios da Recuperação de Evidências Digitais, tópico da seção 2.6, são abortadas descrições do funcionamento e estrutura de algumas tecnologias relacionadas à área. Tal conhecimento teórico é importante na atuação dos especialistas forenses. Assuntos como discos rígidos, discos de estado sólido (SSDs), sistemas de arquivos, e técnicas de *data carving* são detalhados. Também é feito o detalhamento técnico de tecnologias em que a evidência digital pode estar presente. São apresentados técnicas de perícias no Registro e em *logs* do Windows e perícias em dados voláteis.

A seção 2.7, Técnicas Antiforenses e Anti-Antiforenses discute os recursos que geralmente são usados para dificultar as perícias e como essas tecnologias podem ser contornadas. São apresentados tópicos como sanitização de discos, criptografia, quebra de senhas e esteganografia.

Finalmente, na seção 2.8, são abordadas as técnicas de perícias em dispositivos móveis, com ênfase aos que possuam sistema operacional Android.

2.2. Crimes Cibernéticos

Os equipamentos computacionais podem ser utilizados de duas formas para o cometimento de crimes: ferramenta de apoio à prática de delitos convencionais ou alvo/ peça imprescindível da ação criminosa.

Na primeira categoria, os crimes envolvidos são delitos que podem ser cometidos sem o uso de computadores, mas por estarmos cada vez mais envolvidos em um mundo digital, esses crimes tradicionais certamente deixarão vestígios digitais. Analisemos, como exemplo, o crime de corrupção passiva, tão em voga atualmente. Para se corromper, o agente não precisa da ajuda de computadores. Mas, provavelmente suas atitudes ilícitas deixarão rastros digitais: e-mails com parceiros do crime, planilhas e outros documentos digitais que podem materializar o fato criminoso. Dessa forma, praticamente qualquer criminoso pode deixar rastros no mundo digital, tornando a computação forense uma área de muita importância na persecução penal.

A outra categoria são os crimes de informática propriamente ditos, nos quais os computadores são peças imprescindíveis para o cometimento do crime. Sem eles, tais crimes não existiriam. Ataques a *sites*, programas maliciosos para roubo de senhas, programas que sequestram os dados do usuário (*ransomware*), entre outros, são exemplos desse tipo de crime.

Muitos fatos criminosos de repercussão e correlacionados com a computação forense têm surgido na mídia. Abordaremos alguns, como o objetivo de, além de informar, sensibilizar o leitor em relação à dimensão que este tipo de delito tem tomado.

Preso em 2008 pela Polícia Federal, o traficante internacional de drogas Juan Carlos Ramírez Abadía teve seu computador periciado. Segundo matéria jornalística publicada pela Folha de São Paulo, o que causou estranheza à polícia foi ter encontrado centenas de fotos do desenho Hello Kitty, todas enviadas por *e-mail*. Em uma análise mais cuidadosa, descobriu-se que estas imagens carregavam mensagens escondidas pela técnica denominada esteganografia. Entre o conteúdo das mensagens havia ordens para movimentar cocaína entre países e para sumir com pessoas na Colômbia. Uma outra reportagem, do site de notícias G1, traz informações que o grupo extremista Al-Qaeda usou filmes pornográficos para esconder informações de ataques terroristas.

Outro caso de repercussão foi o embate entre o FBI e a Apple relacionado a um iPhone de um suposto terrorista. A polícia estadunidense requisitou que a Apple desenvolvesse uma versão especial do iOS que permitisse que o dispositivo fosse desbloqueado de forma segura. A empresa negou-se a desenvolver qualquer solução tecnológica que comprometesse a segurança de seus dispositivos. A saída encontrada pelo FBI foi pagar um grande quantia a um grupo especializado em segurança da informação que possuía em sua base privada de vulnerabilidades conhecidas uma falha de segurança que permitia desbloquear o iPhone.

Uma forte área de atuação dos cibercriminosos, principalmente no cenário brasileiro, são as fraudes bancárias pela internet. De acordo com estimativas da Febraban – Federação Brasileira de Bancos – 95% das perdas dos bancos do Brasil vem do cibercrime. Em números, o Brasil perde mais de US$ 8 bilhões por ano com fraudes digitais, um dos maiores crimes econômicos no País.

O crime cibernético já é o terceiro que mais causa prejuízo financeiro ao mundo depois do narcotráfico e da falsificação de marcas e de propriedade intelectual.

2.3. Princípios da Computação Forense

A computação forense consiste, basicamente, no uso de métodos científicos para preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação de evidência digital com validade probatória em juízo.

Segundo definição encontrada em Brooks (2014), computação forense é uma disciplina que combina elementos do direito e da computação com objetivo de coletar e analisar dados de sistemas computacionais, redes de computadores, comunicações sem fio e sistemas de armazenamento digitais de tal forma que esses dados sejam válidos na justiça.

Cuidados devem ser tomados para garantir a preservação e coleta dos dados digitais: isolar o local; evitar acessos remotos; utilizar funções de *hash* para garantir a integridade dos dados e a cadeia de custódia.

2.3.1. Preservação e coleta dos dados

Em alguns casos, o perito criminal é chamado para acompanhar uma operação policial em que haja a possibilidade de existir provas digitais. Para esses casos e para outros casos em que a perícia ou a coleta inicial dos dados aconteçam no local onde estão em funcionamento os sistemas computacionais, deve-se tomar precauções para que o local seja adequadamente isolado.

A prova digital pode ser bastante volátil. Se o local não for devidamente isolado, os dados de interesse podem ser corrompidos ou apagados. Para evitar o comprometimento das evidências, recomenda-se não permitir que os usuários dos locais acessem seus computadores, bem como, interromper as comunicações de rede externas para que comandos remotos para limpeza dos dados não possam ser executados.

Ao se deparar com máquinas que estejam desligadas, vide regra, não se deve ligá-las. O principal motivo dessa recomendação é a preservação dos dados. Ao se ligar as máquinas, o próprio processo de inicialização do sistema operacional fará alterações em alguns dados, e essas alterações podem ser detectadas examinando-se os metadados

de carimbo de tempo dos arquivos. Além disso, acessar arquivos de interesse diretamente nas máquinas, também alterará, no mínimo, os dados de tempo dos arquivos. Desse modo, é preferível fazer imagem dos computadores e proceder as análises sobre as imagens.

Os equipamentos, se apreendidos, devem ser etiquetados, constando o nome da pessoa que usava aquele equipamento. Uma recomendação também é perguntar ao usuário a senha de acesso ao dispositivo. Equipamentos com senha podem demandar mais tempo para acesso aos dados ou mesmo inviabilizar a perícia. Dessa forma, não custa nada perguntar a senha. Se o usuário colaborar, anotar a senha para que possa ser usada, caso necessário.

Finalmente, para aqueles dispositivos que tenham conectividade com redes celulares, deve-se colocá-los em modo avião. Não sendo possível, deve-se retirar o chip SIM e desligá-lo. Esse processo é importante, pois os sistemas de dispositivos móveis permitem que os aparelhos sejam bloqueados e os dados apagados remotamente. Colocando-se o dispositivo em modo avião, elimina-se esse risco.

2.3.2. Integridade e cadeia de custódia

Garantir a integridade e prover meios de se assegurar a cadeia de custódia é uma das atividades do *expert* de computação forense.

Segundo Machado (2009), cadeia de custódia é procedimento preponderante e de suma importância para a garantia e transparência na apuração criminal quanto à prova material, sendo relato fiel de todas as ocorrências da evidência, vinculando os fatos e criando um lastro de autenticidade jurídica entre o tipo criminal, autor e vítima.

Na computação forense, o cálculo do *hash* das evidências digitais é um recurso fundamental para a garantia da integridade e da cadeia de custódia da prova. Pelo cálculo e documentação do *hash* da evidência original e da cópia forense, é possível garantir que a cópia é idêntica ao original e que em qualquer momento que se deseje analisar a cópia, basta recalcular o *hash* e verificar se aquela cópia está íntegra.

Toda essa garantia é possível devido as características matemáticas de uma função de *hash*, que é um algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo, obedecendo determinadas propriedades. Essas propriedades asseguram que o resultado do *hash* será praticamente único para aquela coleção de dados original e que qualquer mudança nos dados originais gerará um código de *hash* totalmente diferente.

Do ponto de vista das características técnicas e propriedades necessárias, Stallings (2008) destaca que: uma função *hash* deverá poder ser aplicada sobre um bloco de dados de qualquer tamanho; sempre produzirá uma saída de tamanho fixo;

deverá ser relativamente fácil de se calcular para qualquer bloco de dados, tornando as implementações em *hardware* e *software* práticas; deverá ser resistente à primeira inversão, ou seja, de posse da saída da função deverá ser computacionalmente inviável encontrar o bloco de dados de entrada; deverá possuir resistência fraca a colisões, ou seja, tendo-se o bloco de dados de entrada x, deve ser computacionalmente inviável encontrar um bloco de dados y que gere a mesma saída da função de *hash* e possuir resistência forte a colisões, ou seja, deverá ser computacionalmente inviável encontrar quaisquer pares de blocos x e y cujo resultado da função *hash* seja a mesma.

2.3.3. Análise e apresentação dos resultados

Uma vez obtida as evidências digitais, de forma íntegra e com cuidados para garantir a cadeia de custódia, chega a hora de analisar os dados. É a fase do exame pericial em si. Entre as principais atividades dessa fase, estão, buscar evidências apagadas, buscar determinada evidência em um universo imenso de dados, decodificar e interpretar dados, compreender eventos dos sistemas computacionais envolvidos, entre outras atividades.

Para finalizar todo o trabalho forense, há a redação do laudo pericial, o qual apresentará os resultados da perícia. O principal desavio nesta etapa final é escrever um documento de maneira que seja tecnicamente preciso e compreensível aos operadores do direto.

2.4. Aspectos Jurídicos em Computação Forense

Por ser uma ciência que visa reportar suas análises e resultados a alguma instância da justiça, é estreita sua relação com as leis. Algumas delas afetam diretamente o trabalho dos peritos, pesquisadores e profissionais da área e devem ser de conhecimento desse grupo.

Primeiramente, talvez como a lei fundamental que garante o exame pericial, temos no Código de Processo Penal - Do exame do corpo de delito e das perícias em geral- Art. 158 e159 que dizem:

> Art. 158. Quando a infração deixar vestígios, será **indispensável** o exame de corpo de delito, direto ou indireto, não podendo supri-lo a confissão do acusado.
>
> Art. 159. O exame de corpo de delito e outras perícias serão realizados por perito oficial, portador de diploma de curso superior.

> § 1o Na falta de perito oficial, o exame será realizado por 2 (duas) pessoas idôneas, portadoras de diploma de curso superior preferencialmente na área específica, dentre as que tiverem habilitação técnica relacionada com a natureza do exame.
>
> § 2o Os peritos não oficiais prestarão o compromisso de bem e fielmente desempenhar o encargo.
>
> § 3o Serão facultadas ao Ministério Público, ao assistente de acusação, ao ofendido, ao querelante e ao acusado a **formulação de quesitos e indicação de assistente técnico**.
>
> § 4o O assistente técnico atuará a partir de sua admissão pelo juiz e após a conclusão dos exames e elaboração do laudo pelos peritos oficiais, sendo as partes intimadas desta decisão.

Dessa forma, no âmbito criminal, é obrigatório o exame pericial em todo crime que deixar vestígio. Outro ponto importante, é a possibilidade do perito da defesa, denominado assistente técnico. Esse profissional pode fazer a sua própria análise pericial e apresentar as suas conclusões em relatório próprio para a apreciação do judiciário.

Outra lei importante, com relação próxima a um tipo de perícia em informática, é a que trata do crime de pedofilia. Tipificado no ECA (Estatuto da Criança e do Adolescente), nos artigos 240 e 241.

> Art. 240. **Produzir, reproduzir, dirigir, fotografar, filmar ou registrar**, por qualquer meio, cena de sexo explícito ou pornográfica, envolvendo criança ou adolescente: Pena – reclusão, de 4 (quatro) a 8 (oito) anos, e multa.
>
> Art. 241. **Vender ou expor à venda** fotografia, vídeo ou outro registro que contenha cena de sexo explícito ou pornográfica envolvendo criança ou adolescente: Pena – reclusão, de 4 (quatro) a 8 (oito) anos, e multa.
>
> Art. 241-A. **Oferecer, trocar, disponibilizar, transmitir, distribuir, publicar ou divulgar por qualquer meio, inclusive por meio de sistema de informática ou telemático**, fotografia, vídeo ou outro registro que contenha cena de sexo explícito ou pornográfica envolvendo criança ou adolescente: Pena – reclusão, de 3 (três) a 6 (seis) anos, e multa.
>
> Art. 241-B. **Adquirir, possuir ou armazenar**, por qualquer meio, fotografia, vídeo ou outra forma de registro que contenha cena de sexo explícito ou pornográfica envolvendo criança ou adolescente: Pena – reclusão, de 1 (um) a 4 (quatro) anos, e multa. § 1o A pena é diminuída de 1 (um) a 2/3 (dois terços) se de pequena quantidade o material a que se refere o caput deste artigo.

É importante ressaltar algumas das condutas que são crimes em relação à pedofilia e qual o papel do perito em computação forense nesses casos. Primeiramente,

destaca-se que o simples fato de armazenar, ou seja, possuir fotos de pedofilia no computador ou *smartphone* já é crime. O papel do perito em relação a esse fato é encontrar tais imagens, que podem estar escondidas, apagadas ou criptografadas. Caso essas imagens sejam encontradas, o próximo passo natural é determinar se o proprietário do dispositivo estava compartilhando essas imagens com outros usuários, o que constitui um crime mais grave. Esse compartilhamento pode ocorrer principalmente por aplicativos de redes ponto a ponto (P2P). Cabe ao perito, verificar essa situação e documentar todo o cenário encontrado.

A lei nº 12.965, de 23 de abril de 2014, conhecida como Marco Civil da Internet estipula algumas regras, das quais as que mais interessam à computação forense são as que regulam o armazenamento dos registros de acesso (*logs*) dos usuários, como mostrado a seguir.

> Art. 1o Esta Lei estabelece **princípios, garantias, direitos e deveres para o uso da internet no Brasil** e determina as diretrizes para atuação da União, dos Estados, do Distrito Federal e dos Municípios em relação à matéria.
>
> …
>
> Subseção I
>
> Da Guarda de Registros de Conexão
>
> Art. 13. Na provisão de conexão à internet, cabe ao administrador de sistema autônomo respectivo o dever de **manter os registros de conexão, sob sigilo, em ambiente controlado e de segurança, pelo prazo de 1 (um) ano**, nos termos do regulamento.
>
> § 2o **A autoridade policial ou administrativa ou o Ministério Público poderá requerer cautelarmente que os registros de conexão sejam guardados por prazo superior ao previsto no caput**.
>
> § 5o Em qualquer hipótese, a disponibilização ao requerente dos registros de que trata este artigo deverá ser precedida de **autorização judicial**, conforme disposto na Seção IV deste Capítulo.
>
> Subseção II
>
> Da Guarda de Registros de Acesso a Aplicações de Internet na Provisão de Conexão
>
> Art. 14. Na provisão de conexão, onerosa ou gratuita, é **vedado guardar os registros de acesso a aplicações de internet**.
>
> Subseção III
>
> Da Guarda de Registros de Acesso a Aplicações de Internet na Provisão de Aplicações
>
> Art. 15. O **provedor de aplicações de internet** constituído na forma de pessoa jurídica e que exerça essa atividade de forma organizada, profissionalmente e com

> fins econômicos **deverá manter os respectivos registros de acesso a aplicações de internet, sob sigilo, em ambiente controlado e de segurança, pelo prazo de 6 (seis) meses**, nos termos do regulamento.

A lei nº 12.737, de 30 de novembro de 2012, conhecida como lei Carolina Diekmann, tipifica, ou seja, torna crime, várias condutas relacionadas a atividades de invasão de sistemas de computador, conforme segue.

> Art. 1o Esta Lei dispõe sobre a **tipificação criminal de delitos informáticos** e dá outras providências.
>
> ...
>
> "Invasão de dispositivo informático"
>
> Art. 154-A. **Invadir dispositivo informático alheio, conectado ou não à rede de computadores, mediante violação indevida de mecanismo de segurança e com o fim de obter, adulterar ou destruir dados ou informações sem autorização expressa ou tácita do titular do dispositivo ou instalar vulnerabilidades para obter vantagem ilícita**:
>
> Pena - detenção, de 3 (três) meses a 1 (um) ano, e multa.
>
> § 1o Na mesma pena incorre quem **produz, oferece, distribui, vende ou difunde dispositivo ou programa de computador** com o intuito de permitir a prática da conduta definida no caput.
>
> Art. 154-B. Nos crimes definidos no art. 154-A, **somente se procede mediante representação, salvo se o crime é cometido contra a administração pública** direta ou indireta de qualquer dos Poderes da União, Estados, Distrito Federal ou Municípios ou contra empresas concessionárias de serviços públicos."

Resumidamente, essa lei trata das invasões de sistemas e confecção e uso de software maliciosos (*malware*). É papel do perito, analisar os computadores em que ocorreram as invasões, determinar como elas ocorreram e se possível apontar na direção do responsável por tais crimes.

2.5. Lab de Computação Forense: Preservação e Análise da Prova Digital

O laboratório de computação forense deve possuir *hardware* e *software* especializado que proporcione as condições técnicas, de forma eficiente, para se obter e processar os dados digitais, transformando-os em evidências. Essas tecnologias estão disponíveis em produtos comerciais, *softwares* desenvolvidos por peritos e *softwares* livres.

2.5.1. Duplicação de dados de forma forense

Uma das primeiras atividades a ser realizada no laboratório é a cópia dos dados dos equipamentos originais. É nessas cópias que as análises serão realizadas.

Para realizar uma cópia de forma forense, todos os bits do equipamento original devem ser copiados, inclusive de áreas não alocadas do sistema de arquivo. Além dessa necessidade, a evidência digital deve ser acessada de forma que haja proteção contra escrita na interface em que ela for conectada. Essa precaução é necessária para que, ao se conectar a mídia original, nenhum dado seja alterado. Conectar a mídia original sem proteção de escrita pode alterar dados ou metadados de arquivos e essas alterações podem ser questionadas pelas partes envolvidas. Outra atividade essencial ao fazer a cópia é calcular o *hash* dos dados originais e o da cópia. Esses valores devem coincidir, garantindo-se, com isso, a integridade e a cadeia de custódia das evidências digitais.

Existem equipamentos especializados em duplicação pericial. Esses equipamentos permitem que as cópias sejam feitas de forma bastante simplificada e asseguram as recomendações citadas. Algumas opções de equipamentos que podem existir em um laboratório de computação forense são o Solo IV, da empresa ICS e o Tableaut TD3 da empresa Guidence Software. Esses equipamentos possuem entradas protegidas contra escrita para conexão das evidências originais, diversos tipos de adaptadores para as interfaces mais comuns de mídias de armazenamento, entre elas, adaptadores para conexões IDE, SATA, SAS, USB, cartões de memória SDCard, entre outros. Possuem também a vantagem de serem portáteis, podendo ser levados a campo. As figuras 2.1 e 2.2 ilustram os equipamentos.

Figura 2.1. Solo IV *Fonte da foto: https://portuguese.alibaba.com*

Figura 2.2. Tableaut TD3 *Fonte da foto: http://www.forensiccomputers.com/*

Se usar um equipamento comercial especializado em duplicação de dados não for uma opção, existem soluções de baixo custo para esse processo. Uma forma de realizar essa cópia é usar uma distribuição Linux preparada para análises forenses. Estas distribuições permitem que se monte o disco original do suspeito no modo "somente leitura". Uma vez montado o disco das evidências, as cópias podem ser feitas por programas que acompanham essas distribuições, como, por exemplo, o dd, dc3dd, dcfldd, entre outros. Esses programas farão uma cópia de todos os bits do disco de origem, inclusive áreas não alocadas. Alguns deles já realizarão também o cálculo do *hash* dos dados originais e do arquivo de destino.

Duas distribuições que fornecem ferramental forense são a Deft Linux (http://www.deftlinux.net/) e Caine (http://www.caine-live.net/).

2.5.2. Processamento e Análise dos Dados

Uma fez feita a duplicação pericial dos dados e tendo garantido a sua integridade por meio do *hash*, o próximo passo é o processamento e análise de dados. Essa fase consiste na recuperação dos dados que estão nas mídias, muitos deles apagados, e a disponibilização desses dados aos peritos de modo que possam ser feitas pesquisas sobre eles. Dessa forma, as principais ferramentas dessa etapa do processo deverão entender os sistemas de arquivos envolvidos, executar técnicas de recuperação de dados apagados, indexar esses dados para futuras pesquisas e interpretar essas informações de modo que o grande volume de dados possa ser organizado em subgrupos e tipos para facilitar a análise dos peritos.

Para essa tarefa, os principais softwares comerciais são Encase (https://www.guidancesoftware.com/encase-forensic), FTK (http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk), entre outros. Alternativamente, o IPED (Indexador e Processador de Evidências Digitais) é uma solução desenvolvida por

peritos criminais da Polícia Federal que tem se mostrado bastante interessante e está disponível para o uso de peritos de outras instituições de segurança pública. Por fim, existem as opções livres, como The Sleuth Kit - TSK - e Autopsy (http://www.sleuthkit.org/). Analisaremos esses dois últimos com mais detalhes na próxima seção.

2.5.2.1 – The Sleuth Kit e Autopsy

The Sleuth Kit (TSK) é um conjunto de ferramentas de linha de comando e bibliotecas em C para análise de disco rígidos e recuperação de arquivos. O Autopsy é um ambiente gráfico que proporciona uma interface mais amigável sobre o TSK. Ambas as ferramentas são livres, de código aberto e estão em constante desenvolvimento pelos seus mantenedores.

A importância didática de ferramentas livres é ressaltado por Fagundes, Neukamp e Silva (2011), que apontam que o software de código aberto é uma modelo didático, pois fomenta o pensamento crítico, conta com uma capacidade de adaptação independente, conta com uma comunidade, na qual há compartilhamento de conhecimento e possibilita ao aluno, mesmo fora do ambiente acadêmico, acesso às ferramentas de forma legal.

Segundo Carrier (2006), o TSK é composto por mais de 20 programas, estilo linha de comando, organizados em grupos. Os grupos em que os programas são divididos são baseados nas entidades das estruturas dos sistemas de arquivos. São eles: categoria de sistemas de arquivos, categoria de conteúdo, categoria de metadados, categoria de aplicação e categorias múltiplas.

Pelos comandos do TSK, é possível examinar cada uma das entidades do sistema de arquivos. Para usá-los em sua plenitude, é necessário um entendimento de como os disco rígidos são estruturados e como as estruturas lógicas dos sistemas de arquivos funcionam.

Os programas do TSK são ótimas ferramentas para destrinchar os dados de um disco. Tem um papel didático importante e servem como os blocos de construção para ferramentas mais integradas, porém são pouco eficientes para lidar com diversos casos, nos quais o interesse é a recuperação do maior número de dados possível e a correta visualização deles, em tempo hábil.

Dessa forma surge a necessidade de se utilizar uma ferramenta que integre os diversos programas do TSK e forneça uma interface mais produtiva. Uma opção é o Autopsy.

O Autopsy utiliza as bibliotecas do TSK e apresenta uma interface gráfica intuitiva para o processamento dos dados a serem analisados. Após entrar com alguns dados sobre o caso, deve-se informar o arquivo de imagem, que é a cópia forense

realizado conforme descrito anteriormente. Este arquivo pode estar no formato bruto, também conhecido com *raw* ou dd ou em algum outro formato usado por algum *software* ou equipamento de duplicação de dados. Um formato bastante popular é o formato E01, introduzido pela EnCase e usado por vários outros programas.

As figuras 2.3 e 2.4 ilustram duas telas do *Autopsy*.

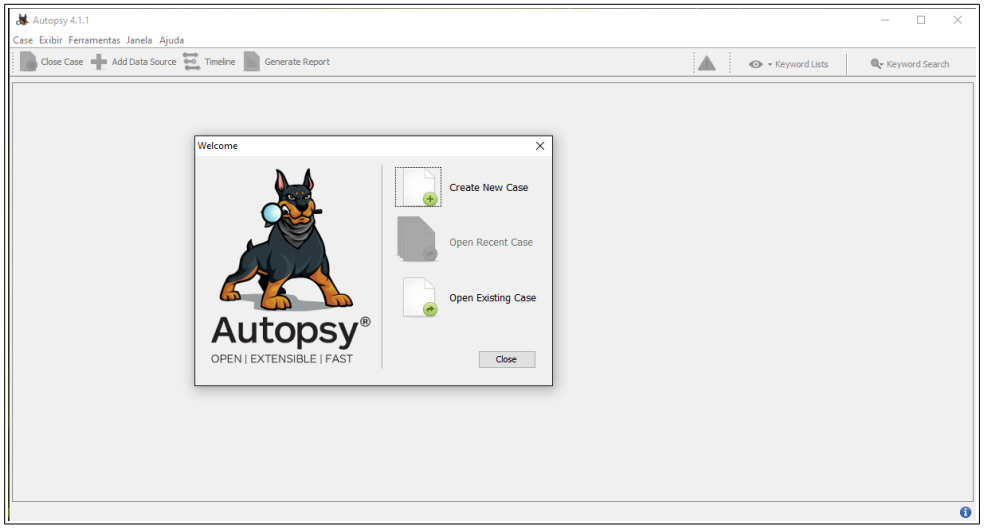


Figura 2.3. Autopsy

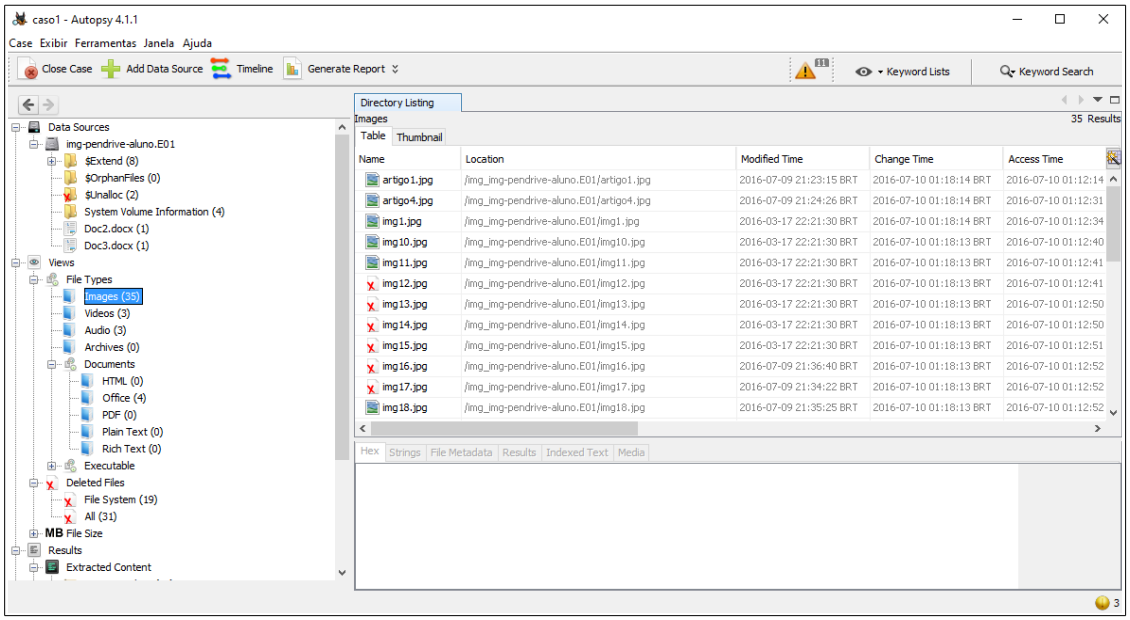


Figura 2.4. Autopsy

2.6. Princípios da Recuperação de Evidências Digitais

Essa seção tem como objetivo apresentar conceitos técnicos que permitam entender como as ferramentas usadas no laboratório de computação forense conseguem chegar aos resultados que se propõem.

Ter o conhecimento técnico do que está sendo feito e não apenas confiar nessas ferramentas e equipamentos como verdadeiras caixas-pretas, que apenas apresentam o resultado, permitirá ao perito uma melhor explanação técnica acerca do que se está periciando, além de subsidiá-lo com conhecimentos suficientes para responder a possíveis questionamentos das partes ou do juízo.

2.6.1. Mídias de Armazenamento

2.6.1.1. Discos rígidos

Os discos rígidos ainda são a principal mídia de armazenamento em massa. São a única parte do computador com componentes mecânicos. Por possuírem peças móveis, são vulneráveis a choques mecânicos. Internamente são compostos por discos magnéticos, nos quais as suas superfícies podem ser magnetizadas para representar bits 0 ou 1. Possuem um desempenho inferior aos componentes eletrônicos do computador, podendo ser um gargalo no desempenho.

Para ler e gravar dados no disco, são usadas cabeças de leitura eletromagnéticas que são presas a um braço móvel, o que permite seu acesso a todo o disco. Para que o disco rígido possa posicionar a cabeça de leitura sobre a área exata referente à trilha que vai ser lida, existem sinais de sincronismo gravados nas superfícies do disco que orientam o posicionamento da cabeça de leitura. Eles são sinais magnéticos especiais, gravados durante a fabricação dos discos, também conhecida como formatação física (Marimoto, 2010).

2.6.1.2. SSDs

Os discos de estado sólido (Solid State Disk – SSD) são memórias de armazenamento permanente. É um tipo de memória *flash*. São memórias eletrônicas que não precisam de alimentação para reter as informações. São constituídas de células compostas por transistores e uma fina camada de óxido de silício que funciona como uma espécie de armadilha para elétrons.

Marimoto (2010) aponta como vantagem dos SSDs o tempo de acesso baixo, com excelentes taxas de leitura e gravação, o que melhora o desempenho consideravelmente em uma grande gama de aplicativos e reduz bastante o tempo de

boot, tornando o sistema muito mais respondível. Os SSDs também oferecem um consumo elétrico mais baixo, são silenciosos, resistentes a impactos e oferecem uma maior segurança contra perda de dados devido a defeitos de hardware, já que não possuem partes móveis.

2.6.2. Sistemas de Arquivos

Um sistema de arquivos é a estrutura lógica utilizada pelo computador para organizar os dados em um meio de armazenamento físico. Ele gerencia procedimentos relacionados a arquivos, tais como, criação, abertura, modificação, remoção etc. Entre os principais, pode-se listar: FAT 12 / 16 / 32, exFAT, NTFS, Ext2, Ext3 e Ext4.

2.6.2.1. FAT

Considerado um dos sistemas de arquivos mais simples. Foi introduzido com o Microsoft DOS e usado como sistema de arquivos padrão de algumas versões do Windows. Ainda é usado em mídias de armazenamento do tipo *flash*, como cartões de memória SDCard e *pendrives*.

Segundo Carrier (2006), um dos motivos de ser considerada simples é possuir um número pequeno de estrutura de dados. As duas principais estruturas são a FAT (*File Allocation Table)* e as entradas de diretório. O conceito de funcionamento básico desse sistema de arquivo é que cada diretório ou arquivo criado aloca uma estrutura de dados denominada entrada de diretório. Nessa estrutura ficam armazenados o nome do arquivo, o tamanho, o endereço do bloco inicial do arquivo, os carimbos de tempo (data de criação, modificação e último acesso) e outros metadados. Se o arquivo for maior do que um bloco, é usada a estrutura de dados FAT para armazenar a sequência de blocos que formam o arquivo.

A figura 2.5 ilustra a entrada de diretórios. A figura 2.6 ilustra a FAT.

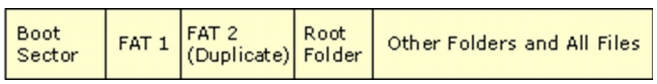


Figura 2.5. Entrada de diretórios FAT

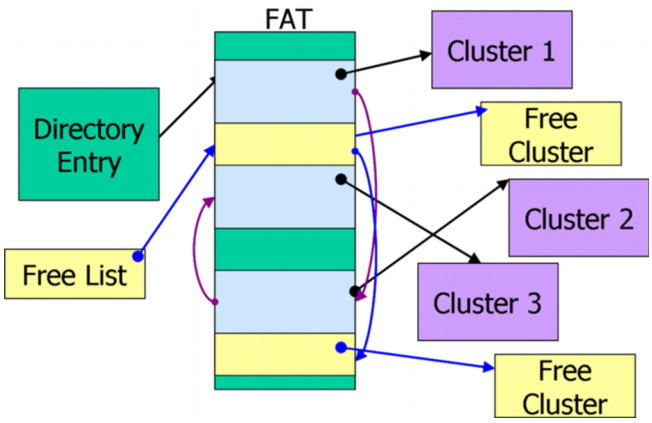


Figura 2.6. FAT

No disco, o sistema de arquivos FAT ocupa três regiões lógicas distintas. A área reservada, que contém informações sobre o sistema de arquivos, a área FAT que contém a estrutura de dados FAT primária e uma cópia de segurança dessa estrutura e a área de dados, onde estarão armazenados os arquivos.

Existem quatro versões do sistema de arquivos FAT.

A FAT 12, usada nos antigos disquetes e discos rígidos "pré-históricos". Permite a utilização de blocos de 512 bytes a 4KB, podendo endereçar de 2MB a 16MB. A FAT 16, usada em disco rígidos muito antigos. Permite blocos de 2KB a 32KB, podendo endereçar 128MB a 2GB. A FAT 32, usada em discos rígidos antigos e nas atuais memórias *flash* (pendrives, cartões de memória etc), permite utilizar blocos de 4KB a 32KB, podendo endereçar 1TB a 2TB. O campo de tamanho de arquivo na FAT é de 32 bits, limitando cada arquivo a no máximo 4GB (2^{32}). Finalmente, a exFAT ou FAT64. Possui limite máximo do tamanho de cada arquivo de 16 Exabytes. Sua capacidade teórica de armazenamento é de 64 ZB (zettabyte), mas a Microsoft não recomenda capacidades acima de 512 TB. Possui suporte para um número maior de arquivos no mesmo diretório (1000). Implementa uma melhor alocação e gerência do espaço livre em disco devido à introdução de uma nova organização da memória (*bitmap*) e possui suporte a lista de controle de acesso.

2.6.2.2. NTFS

O NTFS (*New Technologies File System*) é o atual sistema de arquivos padrão do Windows.

NTFS foi desenvolvido para ser confiável, seguro, com suporte para dispositivos de grande capacidade de armazenamento. Um conceito importante do NTFS é que todos os dados e metadados são armazenados em arquivos. Não existe um layout predefinido

para diferentes áreas do sistema de arquivos, exceto para o setor de *boot*. Um outro recurso adicionado ao NTFS é o *journal*, que permite a recuperação do sistema de arquivos após determinadas falhas. Com o recurso de *journal*, as alterações realizadas no sistema de arquivos são gravadas no arquivo $LogFile. Permite a implementação do conceito de transações, como *redo*, *undo* e *commit*. (Carrier, 2006).

Os metadados são armazenados em arquivos ocultos, no diretório raiz, denominados *metafiles*. Todos eles começam com o caractere "$". A figura 2.7 lista os principais *metafiles*.

A principal estrutura do NTFS é a MFT (*Master File Table*). Ela contém informações sobre todos os arquivos e diretórios. Cada arquivo ou diretório tem, pelo menos, uma entrada na tabela MFT. Uma entrada nessa tabela é composta por um cabeçalho e alguns atributos. Cada tipo de atributo tem uma função própria. Podem ser residentes (atributos pequenos armazenados na própria entrada de diretório da MFT) ou não residentes (o cabeçalho do atributo fica na MFT, mas seu conteúdo fica em blocos do disco rígido). A figura 2.8 apresenta uma lista de alguns atributos.

Entrada MFT	Nome Arquivo	Descrição
0	$MFT	Entrada para a própria MFT.
1	$MFTMirr	Contém backup das primeiras entradas da MFT.
2	$LogFile	Arquivo que armazena o *journal* do NTFS.
3	$Volume	Contém informações do volume, como por exemplo, no nome, identificação e versão do volume.
4	$AttrDef	Contém dados de atributos.
5	.	Contém o diretório raiz do sistema de arquivos.
6	$Bitmap	Contém o estado de alocação de cada bloco do sistema de arquivos.
7	$Boot	Contém o setor de *boot* e o código de *boot*. É o único arquivo do NTFS quem tem a localização estática no disco. Seu conteúdo sempre está no setor 0 do sistema de arquivos.
8	$BadClus	Contém a lista dos blocos que possuem setores defeituosos.
9	$Secure	Contém informações sobre segurança e a lista de controle de acessos do sistema de arquivos.
10	$Upcase	Contém a versão em caixa alta de cada caractere Unicode.
11	$Extend	Diretório que contém arquivos para extensões opcionais.

Figura 2.7: Metafiles

Tipo	Nome	Descrição
16	$STANDARD_ INFORMATION	Informações gerais, como por exemplo, datas de criação, modificação e acesso; proprietário do arquivo; ID de segurança etc.
32	$ATTRIBUTE_LIST	Lista de atributos de arquivos.
48	$FILE_NAME	Nome do arquivo e as últimas datas de acesso, criação e modificação do arquivo.
64	$VOLUME_VERSION	Informações sobre o volume (apenas na versão 1.2 do NTFS).
64	$OBJECT_ID	Identificador único de 16 *bits* de arquivos e diretórios (versão 3.0+ do NTFS).
80	$SECURITY_ DESCRIPTOR	Contém controle de acesso e propriedades de segurança de um arquivo (obsoleto, usado em versões da NTFS anteriores a 3.0).
96	$VOLUME_NAME	Nome do volume.
112	$VOLUME_ INFORMATION	Versão do sistema de arquivos e outros *flags*.
128	$DATA	Conteúdo de arquivo.
144	$INDEX_ROOT	Nó raiz de uma árvore de índices.
160	$INDEX_ALLOCATION	Nós de uma árvore de índices com raiz em $INDEX_ROOT.
176	$BITMAP	Mapa de *bits* para o arquivo $MFT e para os índices.
192	$SYMBOLIC_LINK	Informações de ligações flexíveis (apenas NTFS versão 1.2).
192	$REPARSE_POINT	Informações sobre ponto de *reparse*, que é usado para ligações flexíveis.
208	$EA_INFORMATION	Usado para compatibilidade com OS/2.
224	$EA	Usado para compatibilidade com OS/2.
256	$LOGGED_UTILITY_ STREAM	Contém chaves e informações sobre atributos criptografados (versão 3.0+ do NTFS – *Windows* 2000+).

Figura 2.8. Atributos MFT

Os carimbos de tempo (*timestamps*) do NTFS ficam armazenados nos atributos $STANDARD_INFORMATION e $FILE_NAME. São quatro tipos: data de criação, data de modificação (alterações no $DATA ou $INDEX), data de acesso e data modificação MFT (não visível para usuários do Windows). Os carimbos de tempo são campos de 64 bits, com precisão de nanosegundos.

2.6.2.3. Ext2, Ext3 e Ext4

São os sistemas de arquivos padrão do Linux. Foram projetados para serem rápidos e confiáveis. A cada versão foram adicionadas novas funcionalidades. A principal diferença entre Ext2 e Ext3 é que nesse último foi adicionado suporte a *journal*, com funcionalidade semelhante ao recurso de *journal* discutido no NTFS. A Ext4, versão mais atual dessa família de sistemas de arquivos, adicionou novos recursos, tais como, alocação tardia (*delayed allocation*), carimbos de tempo com maior resolução (nanossegundos), verificação de integridade do *journal* (*journal checksums*), suporte para tamanhos maiores de volumes e arquivos, pré alocação de arquivos e sistemas de verificação mais rápidos. O Ext4 também é utilizado por algumas versões do sistema operacional Android.

As informações sobre o layout básico do sistema de arquivos são armazenados numa estrutura de dados denominada superbloco, que fica armazenada no começo do sistema de arquivos. O conteúdo dos arquivos fica armazenado em estruturas

denominadas blocos, que são agrupamentos de setores consecutivos da mídia de armazenamento. Os metadados de cada arquivo e diretório são armazenados em uma estrutura de dados denominada *i-node*. Os *i-nodes* ficam armazenados na tabela de *i-nodes*. Existem várias tabelas de *i-nodes* distribuídas pelo sistema de arquivo, uma para cada agrupamento de blocos. Os nomes dos arquivos são armazenados em uma estrutura de dados denominada entrada de diretório. Além do nome do arquivo, essa estrutura armazena um ponteiro para o *i-node* relacionado ao arquivo (Carrier, 2006).

A figura 2.9 ilustra as estruturas de dados do sistema de arquivos Ext.

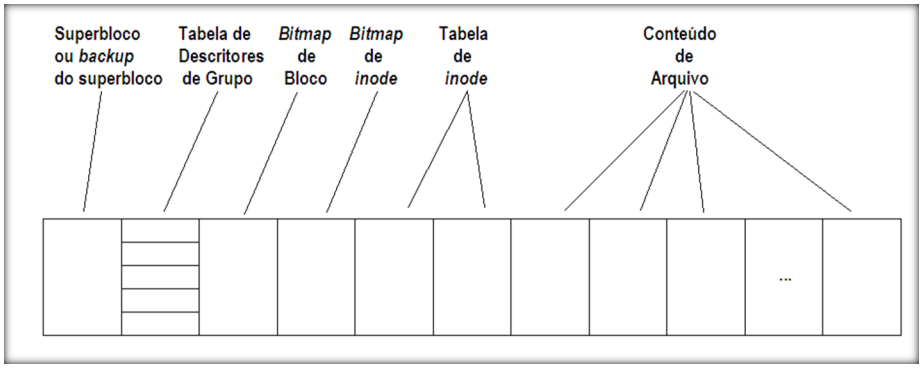


Figura 2.9

O *i-node* armazena diversos metadados dos arquivos, entre eles, permissões, tamanho do arquivo, os carimbos de tempo (*timestamp*) e a lista de blocos que armazenam o conteúdo do arquivo (figura 2.10).

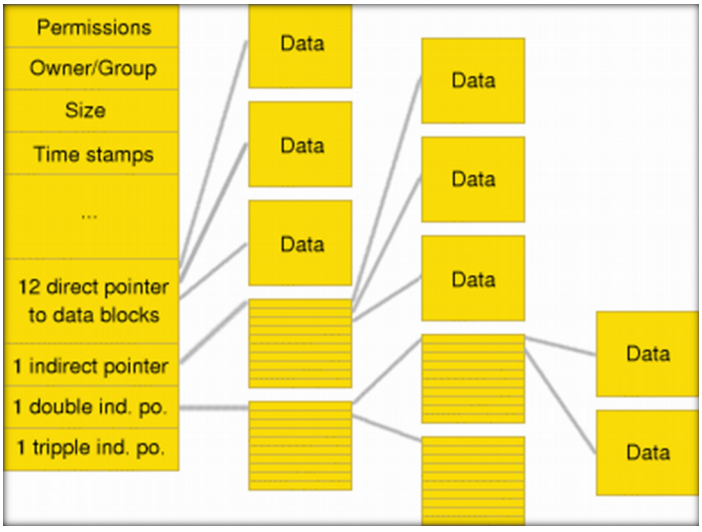


Figura 2.10

2.6.4. Técnicas de Recuperação de Arquivos Apagados

Apesar das peculiaridades de cada sistema de arquivos e das técnicas para recuperá-los, basicamente a recuperação de dados apagados é possível porque ao se apagar um arquivo, ele é apagado apenas logicamente do sistema de arquivos, ou seja, o espaço ocupado por aquele arquivo é liberado para reutilização, mas por questões de desempenho o seu conteúdo permanece intacto até que aquele espaço seja necessário para alocar outro arquivo.

Algumas técnicas de recuperação de dados levam em conta a estrutura de dados providas pelos sistemas de arquivos. Para entender como esse tipo de recuperação de dados é possível, precisamos entender, primeiramente, o que acontece em cada sistema de arquivos quando um arquivo é apagado.

Como mostrado por Carrier (2006), nos sistemas de arquivos FAT ao se apagar um arquivo, o primeiro caractere na tabela de entrada de diretório é substituído por 0xe5 e os endereços na tabela FAT são zerados. Para recuperá-lo, deve-se encontrar o nome dele na tabela de diretório, o endereço do primeiro bloco e os metadados que informam o tamanho do arquivo. De posse da informação de qual é o primeiro bloco e o tamanho do arquivo, a recuperação é trivial. Porém, arquivos fragmentados podem inviabilizar a recuperação pelo uso apenas dessa técnica.

No NTFS, quando um arquivo é apagado, a entrada de diretório na MFT desse arquivo é marcada como não alocada e os blocos desse arquivo são adicionados na tabela de blocos livres. Com isso, a estrutura de alocação de arquivos permanece praticamente intacta, permitindo a recuperação do arquivo até que a entrada de diretório seja reutilizada.

No Ext2, o *i-node* da entrada de diretório é apagado. Para recuperar arquivos apagados, deve-se pesquisar por *i-nodes* não alocados. Encontrando-se um *i-node* não alocado, ele conterá a lista de blocos daquele arquivo apagado. No Ext3 e Ext4, o *i-node* da entrada de diretório não é apagado, porém, os campos com os endereços dos blocos no *i-node* são apagados. Dessa forma, tem-se o *i-node* de determinada entrada de diretório (nome do arquivo), porém não se consegue obter a lista de blocos que compunham esses arquivos. A recuperação de arquivos no Ext3 e Ext4 é mais difícil que no Ext2.

Uma outra técnica promissora de recuperação de arquivos apagados é o *data carving.* No processo clássico de *data carving*, as estruturas do sistema de arquivos não são levadas em consideração.

Merola (2008) cita o exemplo de arquivos PDF e JPEG. Os arquivos PDF possuem uma assinatura inicial, ou seja, começam sempre da mesma forma, o que permite distingui-los de outros tipos de arquivos examinado apenas seu conteúdo. Dessa forma, todos os arquivos PDF iniciarão com os caracteres “%PDF”. Essa assinatura também é conhecida como cabeçalho do arquivo. Alguns arquivos, além do cabeçalho,

possuem também um rodapé, ou seja, sempre terminarão com o mesmo caractere. No caso dos PDFs será "%EOF". Para arquivos JPEG, teremos os padrões "0xFFD8" para o cabeçalho e "0xFFD9" para o rodapé.

É com base nas assinaturas dos arquivos que as técnicas básicas de *data carving* funcionam. Uma ferramenta empregando essa técnica terá uma ampla base de assinaturas dos mais variados tipos de arquivos. Uma vez identificado o início de um arquivo, a ferramenta irá considerando que tudo o que virá depois dessa assinatura é o corpo do arquivo. Ao encontrar o rodapé, a ferramenta conclui a recuperação daquele arquivo e o processo de repete a partir do próximo byte, até que todos os bytes não alocados da mídia de armazenamento sejam processados.

Porém, dificuldades podem ser encontradas nesse processo. Arquivos podem possuir cabeçalho, mas não rodapé. Arquivos podem estar também fragmentados, compactados ou incompletos. Para lidar com essas questões, as técnicas mais avançadas de *data carving* baseiam-se não apenas nas assinaturas dos arquivos, mas também possuem conhecimento das estruturas internas de cada tipo de arquivo, o que permite às ferramentas tentar encaixar todas as peças, num verdadeiro quebra-cabeça de bytes e fragmentos de estruturas de arquivos.

Em relação à recuperação de arquivos nos discos de estado sólido (SSDs), deve-se notar que a dinâmica de leitura e escrita de dados difere dos discos rígidos magnéticos tradicionais, impactando nas técnicas de recuperação de evidências digitais.

Conforme explicado por Gomes (2012), diferentemente dos disco rígidos, nos quais os dados podem ser apagados e sobrescritos de maneira independente, nos SSDs as páginas na memória *flash* não podem ser simplesmente regravadas. Sempre que se precisa gravar dados em uma página já ocupada, a controladora do SSD precisa primeiro apagar os dados anteriores, levando a célula ao seu estado original, para só então, realizar a nova operação de escrita. Além disso, não é possível apagar apenas uma página, deve-se apagar um bloco de páginas. Se houver informações válidas nessas páginas, elas precisam ser copiadas e depois reescritas. Todas essas operações podem comprometer o desempenho do SSD.

Para lidar com essas características, os SSDs utilizam técnicas de coleta de lixo (*garbage collection*). O coletor de lixo será executado em segundo plano, pelo próprio *hardware* do SSD e será responsável por garantir que sempre haja blocos livres, em estado original, prontos para escrita. Para garantir isso, uma de suas tarefas é mover dados, realizando uma espécie de desfragmentação do disco. Essa característica tem uma impacto negativo sobre a recuperação de arquivos apagados, tendo em vista que a chance de sobreposição de dados não alocados é bem maior por conta do coletor de lixo.

2.6.5. Perícias em Dados Voláteis

Informações preciosas podem estar armazenadas apenas na memória RAM. Se o conteúdo do disco rígido estiver criptografado, fazer a extração e análise dos dados voláteis pode possibilitar a obtenção da chave usada para proteger os dados do disco. Outras informações como processos em execução e bibliotecas de software carregadas também podem ser obtidas por meio desse tipo de análise.

Silva e Lorens (2009) discorrem sobre a necessidade de um exame pericial em memória RAM, também conhecido como *live forensics*, tendo em vista que circunstâncias específicas justificam a realização de procedimentos de coleta de vestígios digitais no local em que se encontram instalados os equipamentos computacionais, enquanto ligados e em funcionamento normal. Instalações de equipamentos de grande porte, não convencionais, ou que suscitem o risco de perda de informações significativas ou ainda, a inviabilização da perícia são exemplos dessas circunstâncias. Destaca-se, também, a situação cada vez mais frequente do uso de criptografia nas mídias de armazenamento.

A primeira tarefa a ser realizada em uma perícia de dados voláteis é obter uma cópia da memória RAM. O termo *dump* de memória também é usado para se referir a este tipo de cópia. Existem várias ferramentas que podem ser usadas para essa tarefa. É interessante que essa ferramenta possa ser executada na máquina alvo sem a necessidade de instalação, para não escrever no disco e correr o risco de sobrescrever algum dado não alocado. Um exemplo de ferramenta livre para Windows que faz a cópia de memória é o *FTK Imager Lite* (figura 2.11).

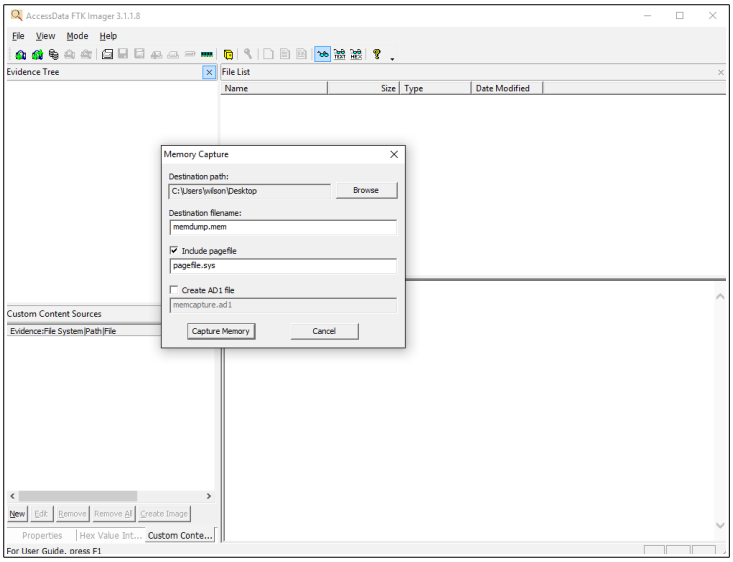


Figura 2.11

Obtida a cópia da memória RAM, é preciso saber interpretá-la. Para essa tarefa existem softwares que podem auxiliar o perito. Um deles é o *framework* livre *Volatility (*http://www.volatilityfoundation.org/*)* (figura 2.12).

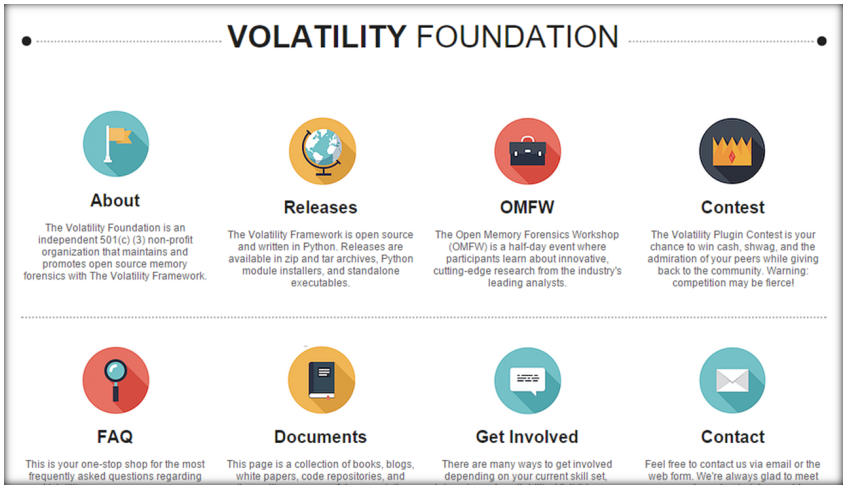


Figura 2.12

O *Volatility* é um conjunto de ferramentas abertas, escritas em *Python*, destinado à extração de conteúdos digitais armazenados em memória volátil de sistemas operacionais Windows. Realiza interpretação (*parser*) de *dump* de memória, *crash dump*, arquivo de hibernação, *snapshot* de máquinas virtuais etc.

Com o uso desse *framework,* podem ser obtidos dados referentes a processos em execução, soquetes de rede abertos, DLLs carregadas para cada processo, arquivos abertos para cada processo, chaves de registro para cada processo, memória endereçável de um processo, módulos do *kernel* do sistema operacional, chaves criptográficas, entre outros.

2.6.6. Busca de evidências no Registro do Windows

O registro do Windows é um banco de dados hierárquico que armazena uma grande quantidade de informações sobre a configuração do Windows, aplicações instaladas e informações sobre atividades dos usuários que interagiram com o sistema operacional. Boa parte dessa informação pode ser de interesse da perícia computacional forense.

Em Carvey (2009), é discutida a importância da análise do Registro em caso de softwares maliciosos. Segundo o autor, softwares maliciosos podem deixar rastros digitais no Registro e interpretar tais rastros dará, ao analista, pistas sobre o comportamento do programa malicioso.

As informações do Registro são organizadas de forma hierárquica. No nível mais alto, existem cinco chaves principais ou raízes. A figura 2.13 descreve essas chaves.

Chave raiz	Descrição
HKEY_CURRENT_USER*	Armazena informações, *profiles* e preferências do usuário que está localmente conectado ao sistema no momento. O *profile* do usuário fica localizado em \Documents and Settings\Nome Usuário\Ntuser.dat, e a partir do *Windows Vista*, em \Users\Nome Usuário\Ntuser.dat.
HKEY_USERS	Armazena informações sobre todas as contas de usuários do sistema.
HKEY_CLASS_ROOT*	Armazena as associações de arquivos e informações de registro dos objetos COM (*Component Object Model*).
HKEY_LOCAL_MACHINE	Armazena a maior parte das configurações do sistema operacional. Exemplos de sub-chaves: BCD (*Boot Configuration Database* – apenas a partir o *Windows Vista*), COMPONENTS, HARDWARE, SAM, SECURITY, SOFTWARE e SYSTEM.
HKEY_CURRENT_CONFIG*	Armazena informações sobre a configuração atual de hardware.
HKEY_PERFORMANCE_ DATA	Armazena informações sobre o desempenho do sistema. (Só é possível acessar essa chave por meio das APIs de programação do *Windows*).

Figura 2.13

* Essas chaves são, na verdade, links para outras chaves que ficam armazenadas embaixo das chaves raízes que não são links. Devido à importância delas, a Microsoft adicionou os links ao nível raiz.

Cada chave pode ter associado um tipo de valor. Os tipos de valores possíveis estão descritos na figura 2.14.

Tipo	Descrição
REG_NONE	Nenhum valor.
REG_SZ	String Unicode de tamanho fixo.
REG_EXPAND_SZ	String Unicode de tamanho variável que pode ter variáveis de ambiente.
REG_BINARY	Dados binários.
REG_DWORD	Número de 32 *bits*.
REG_DWORD_ LITTLE_ENDIAN	Número de 32 *bits* com bytes menos significativos primeiro.
REG_DWORD_BIG_ ENDIAN	Número de 32 *bits* com bytes mais significativos primeiro.
REG_LINK	Ligação simbólica Unicode.
REG_MULTI_SZ	Arranjo de strings Unicode terminadas em zero.
REG_RESOURCE_LIST	Descrição de recursos de hardware.
REG_FULL_RESOURCE_ DESCRIPTOR	Descrição de recursos de hardware.
REG_RESOURCE_ REQUIREMENTS_LIST	Lista de recursos.
REG_QWORD	Número de 64 *bits*.
REG_QWORD_LITTLE_ ENDIAN	Número de 64 *bits* com bytes menos significativos primeiro.

Figura 2.14

Fisicamente, os dados do Registro ficam armazenados em arquivos denominados *HIVE*. A descrição desses arquivos é apresentada na figura 2.15.

Chave do Registro	Arquivo *Hive*
HKEY_LOCAL_MACHINE\System	\Windows\System32\config\System
HKEY_LOCAL_MACHINE\SAM	\Windows\System32\config\SAM
HKEY_LOCAL_MACHINE\Security	\Windows\System32\config\Security
HKEY_LOCAL_MACHINE\Software	\Windows\System32\config\Software
HKEY_LOCAL_MACHINE\Hardware	Chave volátil, armazenada apenas na RAM.
HKEY_LOCAL_MACHINE\System\ Clone	Chave volátil, armazenada apenas na RAM.
HKEY_USERS\[SID Usuário]	\Documents and Settings\[usuário]\NTUSER.DAT (até Windows XP) \Users\[usuário]\NTUSER.DAT (a partir do Windows Vista)
HKEY_USERS\Default	\Windows\System32\config\Default

Figura 2.15

O Registro é uma grande fonte de informação, mas para ser útil ao perito, esses dados devem ser extraídos e interpretados. Fazer essa atividade sem ajuda de alguma ferramenta é contra producente. Uma opção de automatizar esse processo de extração, interpretação e apresentação desses dados é a ferramenta RegRipper.

O RegRipper é um *framework* composto por uma coleção de *scripts* escritos na linguagem Perl. Os *scripts* funcionam como *plug-ins* do *framework*. Novos *scripts* podem ser adicionados ou escritos por terceiros.

O RegRipper lê as informações do Registro diretamente dos arquivos (*hives*) que as armazenam, interpreta esses dados e os disponibiliza para o usuário.

A seguir são apresentadas algumas das informações que podem ser obtidas do Registros, e os comandos do RegRipper para obtê-las.

Nome do computador:

Informação encontrada na chave SYSTEM\CurrentControlSet\Control\ ComputerName\ActiveComputerName.

```
C:\RegRipper>rip -p compname -r f:\T\system
Launching compname v.20080324
ComputerName = COMPUTADORNTFS
```

Informações relacionadas à versão do sistema operacional:

Informações encontradas nas chaves: SYSTEM\ControlSet00x\Control\Windows e SOFTWARE\Microsoft\Windows NT\CurrentVersion

```
C:\RegRipper>rip -p winnt_cv -r f:\T\software
Launching winnt_cv v.20080609
WinNT_CV
Microsoft\Windows NT\CurrentVersion
LastWrite Time Wed Nov 18 09:25:43 2009 (UTC)
SubVersionNumber :
  RegDone :
  RegisteredOrganization : .
  RegisteredOwner : .
  CurrentVersion : 5.1
  CurrentBuildNumber : 2600
SoftwareType : SYSTEM
SourcePath : F:\I386
SystemRoot : C:\WINDOWS
PathName : C:\WINDOWS
CSDVersion : Service Pack 2
CurrentType : Multiprocessor Free
ProductName : Microsoft Windows XP
ProductId : 55274-640-8816093-23950
BuildLab : 2600.xpsp_sp2_rtm.040803-2158
InstallDate : Mon Oct 26 15:26:01 2009 (UTC)
CurrentBuild : 1.511.1 () (Obsolete data - do not use)
```

Ressalta-se que o campo “LastWrite Time Wed Nov 18 09:25:43 2009 (UTC)” é a data e horário em que o sistema foi desligado (shutdown) pela última vez.

Interfaces de rede e endereço IP

Informação encontrada na chave ControlSet00x\Services\Tcpip\Parameters\Interfaces.

```
C:\RegRipper>rip -p networkcards -r f:\T\software
Launching networkcards v.20080325
NetworkCards
Microsoft\Windows NT\CurrentVersion\NetworkCards
AMD PCNET Family PCI Ethernet Adapter

C:\RegRipper>rip -p nic_mst2 -r f:\T\system
Launching nic_mst2 v.20080324
Network key
ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}

ControlSet001\Services\Tcpip\Parameters\Interfaces
LastWrite time Mon Oct 26 15:20:28 2009 (UTC)

Interface {A69C78C2-98A2-4F6A-9FEC-534A380240B1}
Name: Conex ├úo local
Control\Network key LastWrite time Mon Oct 26 15:20:31 2009 (UTC)
Services\Tcpip key LastWrite time Fri Nov 27 14:20:03 2009 (UTC)
    DhcpDomain     = localdomain
    DhcpIPAddress  = 192.168.145.131
    DhcpSubnetMask = 255.255.255.0
    DhcpNameServer = 192.168.145.1
    DhcpServer     = 192.168.145.254
```

Outras informações que podem ser obtidas do Registro são *Wireless* SSIDs, lista de dispositivos móveis que foram conectados a USB, contas do usuário no sistema, atividades do usuário etc.

2.6.7. Busca de Evidências nos LOGs do Windows

Assim como o Registro do Windows, os *logs* do sistema são uma fonte de dados a ser analisada em determinados tipos de perícia.

Vários tipos de eventos do sistema operacional e atividades de programas e usuários são registrados nos *logs*. Existem quatro categorias padrão, que os dividem conforme seu tipo: aplicação, segurança, instalação e sistema. A figura 2.16 ilustra o programa Visualizador de Eventos do Windows, no qual é possível ter acesso ao conteúdo dos *logs* e fazer pesquisas simples.

Caso haja necessidade de se fazer pesquisas mais elaboradas, pode-se usar a ferramenta de linha de comando LogParser. Com essa ferramenta é possível fazer pesquisas nos *logs* usando-se uma sintaxe similar as pesquisas realizadas em linguagem SQL. O LogParser é ilustrado na figura 2.17.

Fisicamente os *logs* ficam armazenados em arquivos específicos. No Windows XP, ficam armazenados na pasta “%SystemRoot%\System32\Config”, nos arquivos sysevt.evt, secevent.evt, appevent.evt, entre outros. Do Windows 7 em diante ficam armazenados nas pastas “%SystemRoot%\System32\winevt\Logs”, nos arquivos Application.evtx, Security.evtx, System.evtx, entre outros.

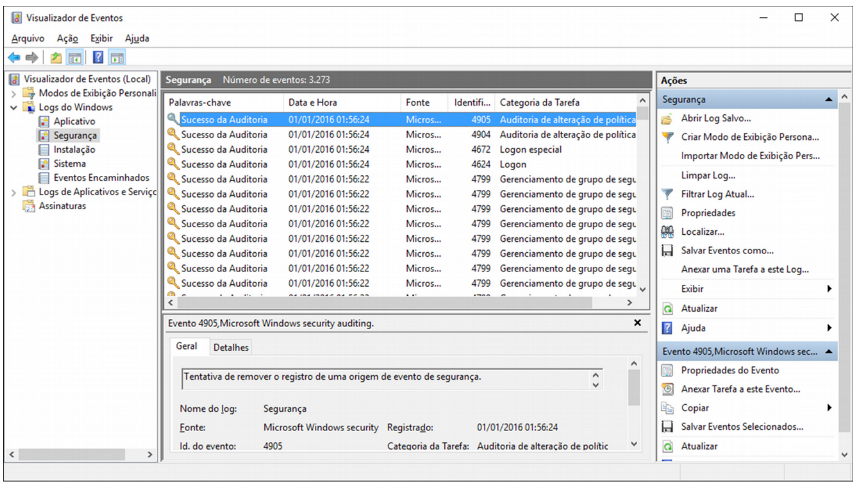


Figura 2.16. Visualizador de Eventos do Windows

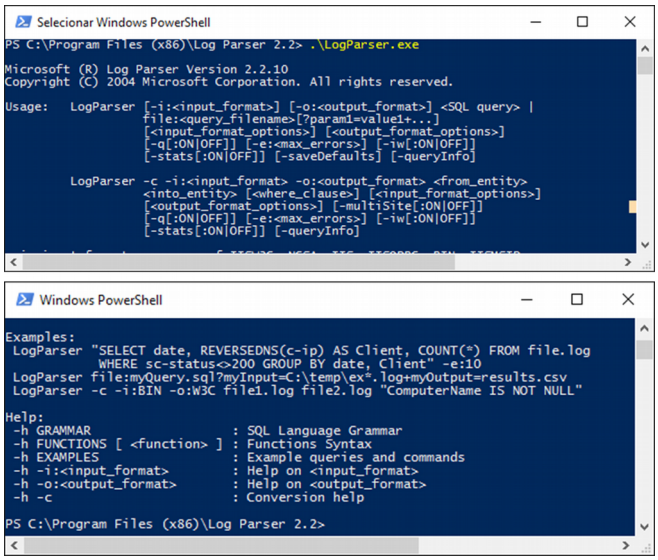


Figura 2.17

2.7. Técnicas Antiforenses e Anti-Antiforenses

Em uma definição de técnicas antiforenses encontrada em Velho et al (2016), os autores classificam-na como um conjunto de técnicas que objetivam inviabilizar, dificultar, iludir ou impossibilitar que a análise forense ocorra de forma satisfatória.

Basicamente, trata-se de formas de manipular os dados digitais de tal forma que esses dados sejam destruídos de maneira irrecuperável, ou, de alguma forma, não possam ser acessados pelo perito, ou ainda, que iludam o perito em suas conclusões.

Portanto, cabe ao perito conhecer sobre essas técnicas, saber identificá-las e contorná-las sempre que possível. No restante dessa seção são apresentadas as principais técnicas antiforenses e possíveis formas de lidar com elas.

2.7.1. *Wipe* ou Sanitarização de Dados

Como foi apresentado na seção 2.6, ao se apagar um arquivo, os dados desse arquivo não são realmente apagados, são feitas algumas alterações nas estruturas de controle de alocação de arquivos e aquele espaço ocupado pelo arquivo fica disponível para ser reutilizado, mas principalmente por motivo de desempenho, os dados desse arquivo apagado permanecem na mídia de armazenamento, até o momento em que forem reutilizados por outro arquivo. A partir desse momento, quando os dados são sobrescritos por um arquivo novo, a recuperação torna-se inviável.

É por isso que existe uma forma de apagar um arquivo de forma irrecuperável. Para isso, além de ser marcado nas estruturas do sistema operacional como apagado, o seu conteúdo em todo o disco deve ser sobrescrito. Existem inclusive protocolos para realizar essa técnica, conhecida como *wipe* ou sanitização de dados. Dependendo da sensibilidade e importância dos arquivos a serem apagados, esses protocolos recomendam que a área da mídia de armazenamento em que os dados estavam armazenados seja sobrescrita diversas vezes. A figura 2.18 ilustra o programa Disk Wipe, que entrega a técnica de sanitização de dados em uma mídia completa. Nota-se que se pode escolher qual protocolo de *wipe* utilizar. Na figura, são apresentados de cima para baixo os protocolos do menos para o mais seguro.

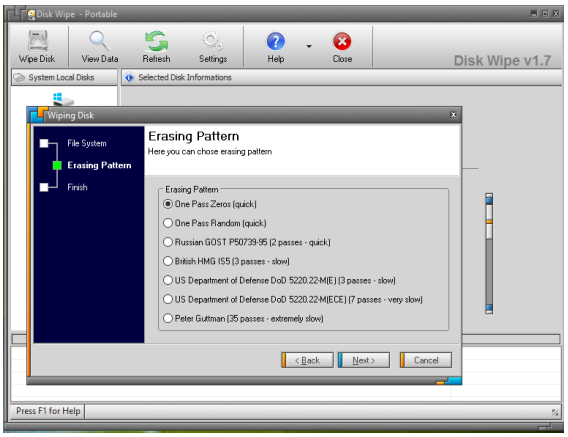


Figura 2.18. Disk Wipe

2.7.2. Criptografia e Quebra de Senhas

Criptografia passa a ser cada vez mais popularizada, sendo que vários sistemas já estão sendo configurados por padrão com seus dados criptografados. Há também diversas softwares que podem ser usados para criptografar arquivos, partes de uma mídia de armazenamento ou a mídia inteira.

Esses recursos, importantes para a segurança do usuário, representam um desafio técnico para os peritos, que precisam tentar ao máximo encontrar evidências digitais, mesmo que estas estejam protegidas por criptografia.

Desse cenário pericial, surge a necessidade de técnicas de quebra de criptografia e senhas, tornando essa área mais uma área da computação que deve ser dominada pelos peritos.

Mas como quebrar a criptografia, tendo em vista que a maioria dos sistemas utilizam criptografia forte, com algoritmos robustos e chaves grandes? A resposta encontra-se em atacar o elo mais fraco da segurança da informação, o usuário. Segundo

Velho et al (2016), pesquisas mostram que a maioria dos usuários usa senhas fracas. A capacidade humana de memorização não facilita que usuários guardem como senhas sequências muitos grandes e aleatórias. Geralmente serão usadas expressões que são familiares aos usuários e as senhas tendem a se repetir em diversos sistemas.

Com isso, ao se deparar com conteúdo criptografado, o perito deve ao menos tentar as técnicas básicas de recuperação de senhas, limitando-se aos recursos computacionais e a um prazo de tempo de tentativa estipulado.

2.7.2.1 Ataques a Dados Criptografados

Podemos definir os ataques a sistemas com senha em dois tipos. Os ataques *on-line*, que visam sistemas que estão em funcionamento no momento dos ataques. Esse tipo de ataque é menos promissor, já que o tempo de resposta em que várias senhas podem ser testadas é alto.

Os ataques *offline* ou *post-mortem*, tentam decifrar os dados já obtidos das mídias de armazenamento, mas que ainda não estão acessíveis por estarem criptografados. É um ataque mais promissor que o anterior, pois a taxa de senhas que podem ser testadas é muito superior. Para a computação forense, esse é o tipo de ataque que mais interessa e é esse tipo que será discutido no restante da seção.

Para se quebrar a criptografia no ataque *offline* deve-se descobrir qual foi a senha usada pelo usuário para criptografar os dados. Para tanto, as possíveis senhas são testadas uma a uma. Porém, essa tarefa computacional é altamente paralelizável. Dessa forma, quanto mais processadores o perito tiver a disposição para a tarefa, mais rápido ela poderá ser cumprida. Atualmente, as duas formas mais utilizadas para paralelizar essa tarefa é por meio de *cluster* de computadores ou por meio de placas de processamento gráfico (GPUs).

Na opção de *cluster* de computadores, usa-se várias máquinas trabalhando em paralelo e em cooperação para o processamento dos ataques ao conteúdo criptografado. Geralmente usa-se um esquema em que uma das máquinas é um ponto central que gerencia todas as demais, distribuindo a carga de processamento.

Outra forma de conseguir alto nível de paralelização é por meio do uso de GPUs. As GPUs são projetadas com diversos núcleos de processamento para atividades específicas. Essa arquitetura pode ser usada para paralelizar as computações necessárias para quebra de senhas. Uma GPU voltada para jogos, pode ter até aproximadamente 3.000 núcleos (*cores*). A figura 2.19 ilustra uma comparação da arquitetura *multicore* de uma CPU e de uma GPU.

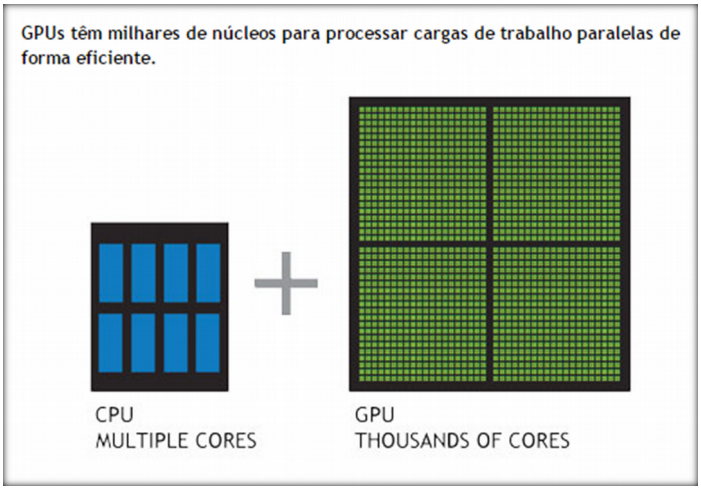


Figura 2.19. Arquitetura *multicore* CPU x GPU

A figura 2.20 exibe um teste de desempenho de quatro configurações de máquinas usando GPUs processando quebra de criptografia em vários algoritmos.

PC1: Windows 7, 32 bit · Catalyst 14.9 · 1x AMD hd7970 · 1000mhz core clock · oclHashcat v1.35

PC2: Windows 7, 64 bit · ForceWare 347.52 · 1x NVidia gtx580 · stock core clock · oclHashcat v1.35

PC3: Ubuntu 14.04, 64 bit · ForceWare 346.29 · 8x NVidia Titan X · stock core clock · oclHashcat v1.36

PC4: Ubuntu 14.04, 64 bit · Catalyst 14.9 · 8x AMD R9 290X · stock core clock · oclHashcat v1.35

Hash Type	PC1	PC2	PC3	PC4
MD5	8581 Mh/s	2753 Mh/s	115840 Mh/s	92672 Mh/s
SHA1	3037 Mh/s	655 Mh/s	37336 Mh/s	31552 Mh/s
SHA256	1122 Mh/s	355 Mh/s	14416 Mh/s	12288 Mh/s
SHA512	414 Mh/s	104 Mh/s	4976 Mh/s	4552 Mh/s
SHA-3(Keccak)	179 Mh/s	92 Mh/s	3400 Mh/s	2032 Mh/s
RipeMD160	1810 Mh/s	623 Mh/s	23936 Mh/s	20016 Mh/s
Whirlpool	65845 kh/s	85383 kh/s	1480000 kh/s	1122304 kh/s
LM	1388 Mh/s	450 Mh/s	15616 Mh/s	16392 Mh/s
NTLM	16916 Mh/s	4185 Mh/s	250360 Mh/s	175808 Mh/s
NetNTLMv1	9108 Mh/s	2330 Mh/s	56448 Mh/s	97800 Mh/s
NetNTLMv2	589 Mh/s	200 Mh/s	7944 Mh/s	6496 Mh/s
WPA/WPA2	142 kh/s	48 kh/s	2096 kh/s	1536 kh/s

Figura *2.20.* Desempenho hashcat Fonte: www.hashcat.com, acesso em 22/09/2016

Mesmo com todo poder de processamento dos *clusters* e das GPUs, testar todas as combinações, num ataque denominado força bruta, não é viável para senhas com um tamanho e complexidade razoáveis (mais de 8 caracteres começa a inviabilizar a força bruta).

Por conta disso, deve-se tentar antes ataques mais inteligentes, nos quais a chance de se descobrir a senha seja melhor do que o acaso. O ataque de força bruta deve ser o último a ser utilizado, apenas quando todos os outros tiverem falhado.

Um dos ataques que pode obter sucesso é o ataque de dicionário. Nesse ataque, tenta-se todas as senhas de um determinado dicionário (lista de palavras), como por exemplo, um dicionário com todas as palavras da língua portuguesa. É um ataque rápido de ser realizado. Nesse tipo de ataque, a criatividade é o limite. Pode-se tentar dicionários temáticos, palavras que sejam da lista de interesses do alvo e até mesmo buscar dicionários de dados recuperando todas as palavras de outros dispositivos de informática do alvo que não estejam criptografados. Não obtendo-se sucesso, pode-se tentar a abordagem híbrida. Nesse ataque, cada palavra do dicionário sofrerá determinada variação, como por exemplo, colocar o primeiro caractere em maiúsculo, adicionar números ao final de cada palavra etc. Novamente, a criatividade é o limite. Vale ressaltar que, quanto mais variedade for adicionada, maior será o tempo necessário para completar o ataque.

2.7.3. Esteganografia e Esteganálise

Esteganografia, ou escrita oculta, pode ser usada como uma técnica antiforense. Segundo Velho et al (2016), esteganografia é o estudo de técnicas para ocultar a existência de uma mensagem dentro de outra, uma forma de segurança por obscurantismo. Ainda segundo esses autores, a informação a ser escondida é inserida em um arquivo hospedeiro, que precisará ser capaz de sofrer pequenas alterações em seus bytes e, ainda assim, reter suas principais características.

A esteganálise tem como objetivo descobrir e revelar mensagens ocultas por técnicas esteganográficas. Sua base é a análise estatística, buscando padrões de ocorrência do uso de técnicas de esteganografia nas mídias suspeitas.

2.8. Perícias Em Dispositivos Móveis

A popularização dos dispositivos móveis tem os tornados fonte de dados imprescindível na busca de evidências digitais. Diferente da perícia em computadores, na qual é possível retirar o disco rígido e fazer uma cópia bit a bit de seu conteúdo em ambiente controlado, as extrações de dados em dispositivos móveis devem ser feitas usando as próprias interfaces desses dispositivos. Essa característica impõe alguns desafios técnicos. Muitas vezes, mecanismos de segurança desses dispositivos têm que ser ultrapassados para se chegar à informação que se quer extrair, tornando esse tipo de perícia mais invasiva e os riscos de comprometimento das informações e do próprio dispositivo, maiores.

Cada tipo de dispositivo móvel tem suas particularidades, sendo que o mercado atual é dominado por dispositivos com sistema operacional Android e dispositivos da Apple. Além dos aspectos gerais, válidos para qualquer fabricante, o restante da seção focará em dispositivos com o sistema Android.

Em relação à tecnologia disponível de extração de dados de celulares e afins, temos um cenário parecido com o já apresentado anteriormente, ou seja, existem soluções comerciais e soluções livre e/ou de código aberto. As soluções comerciais oferecem uma interface intuitiva para a extração dos dados e constantes atualizações conforme a evolução dos sistemas operacionais e aplicativos dos dispositivos móveis. Entre os principais produtos, podemos citar Microsystemation XRY e Cellebrite UFED, ilustrados nas figuras 2.21 e 2.22, respectivamente.

Figura 2.21. XRY *Fonte da foto: http://aresources.pt/*

Figura 2.22 Cellebrite UFED *Fonte da foto: http://www.tecmundo.com.br/*

Quando falamos em extração de dados de dispositivos móveis temos, basicamente, seis tipos prenominantes: extração manual, extração lógica, extração física, JTAG, chip-off e micro read. As três últimas necessitam de conhecimento e manipulação do hardware e fogem do escopo deste trabalho.

Segundo apresentado por Velho et al (2016) e Tamma e Tindall (2015), cada tipo de extração apresenta a sua particularidade. A extração manual é a mais simples de todas e consiste em acessar o dispositivo manualmente e transcrever ou fotografar o conteúdo visível pelo próprio dispositivo. É um tipo de extração demorada, não recupera informações apagadas, devendo ser feita apenas em último caso, quando nenhum outro tipo de extração funcionou. Na extração lógica, o dispositivo móvel é conectado ao computador por meio de cabo USB ou rede sem fio de curto alcance (*bluetooth*, por exemplo) e o software de extração usa as APIs do sistema operacional para extrair os arquivos. Esse tipo extração não recuperará arquivos apagados. Uma forma especial de extração lógica é a extração sistema de arquivos, que usará um conjunto de APIs do sistema de arquivos. Essa extração poderá recuperar dados apagados que estejam em arquivos tipo banco de dados, como por exemplo, arquivos do banco de dados SQLite. Na extração física, será feita uma cópia bit a bit da memória interna do dispositivo. É o tipo que permite a maior recuperação de dados. Pode ser feita por meio de um *bootloader* específico, copiado para o dispositivo móvel ou por meio da instalação de um programa no dispositivo que copiará toda a memória. Essa segunda opção é mais invasiva e necessitará de privilégios de super-usuário (*root*).

2.8.1. Perícias em Dispositivos Android

Nessa seção serão apresentadas algumas características do sistema operacional Android, os principais locais em que os dados ficam armazenados e formas de se extrair dados dos dispositivos usando os próprios recursos do Android ou usando software livre.

O Android foi desenvolvido baseado no *kernel* do Linux, com modificações para adequar o sistema ao ambiente de dispositivos móveis. Uma vez obtido acesso ao dispositivo, muitos comandos do Linux podem ser executados no Android, inclusive aqueles para navegação pelos diretórios e para realizar cópias de dados.

Um dos aspectos importantes que o especialista deve ter conhecimento em relação a esse tipo de perícia é a segurança implementada pelo Android. A segurança em dispositivos móveis é um recurso importante aos usuários, mas pode dificultar o trabalho do perito. Segundo Tamma e Tindall (2015), recursos importantes de segurança são assegurados utilizando-se os recursos de segurança do *kernel* do Linux. Por essa implementação, é garantido um modelo de acesso baseado em permissões, isolamento de processos e um mecanismo de segurança para chamadas interprocessos (IPC).

No modelo de segurança do Android, cada aplicação é executada com seu próprio UID, não permitindo com isso que, vide regra, uma aplicação acesse os dados

armazenados no sistema de arquivos de outra. As aplicações também são executadas em uma *sandbox*, rodando em uma máquina virtual Java modificada, chamada Dalvik. Além disso, cada aplicação deve ser assinada por uma chave criptográfica do fornecedor da aplicação e deve declarar em um arquivo de manifesto quais são os recursos do sistema que ela necessita.

2.8.1.1. Processo de *Boot* do Android

Entender o processo de *boot* do Android ajudará na compreensão de como algumas técnicas de extração de dados funcionam.

O processo de *boot* é dividido em seis fases: código de *boot* ROM, o *boot loader*, o *kernel*, processo init, o zygote e Dalvik e o serviço de sistema. A primeira fase é o *boot* ROM, que inicializa o hardware e procura por uma mídia de *boot*. A próxima fase é o *boot loader*, responsável por carregar o sistema operacional. É o *boot loader* que também permite inicializar o dispositivo em outros modos, como o modo *fastboot*, *recovery* etc. Numa inicialização padrão, o *boot loader* carrega o *kernel* na memória e passa o controle para ele. Uma vez carregado, o kernel monta o sistema de arquivos de *root* (*rootfs*) e executa o primeiro processo do sistema, o init. O processo init executará os comandos do script init.rc. O processo zygote, responsável por criar as máquinas virtuais Java Dalvik será iniciado. Por fim, o serviço do sistema inicializá diversos serviços e o dispositivo Android estará operacional para o usuário.

2.8.1.2. Acessando Dispositivos Android por meio do ADB

O ADB (*Android Debug Bridge*) é um mecanismo de depuração de aplicações Android que permite que o dispositivo seja acessado de um computador. Pode ser uma das formas de se obter os dados do dispositivo.

Para usar esse recurso, o ADB deve estar instalado no computador do perito, por meio da instalação do pacote de desenvolvimento Android SDK (https://developer.android.com/studio/index.html). No dispositivo móvel, o software necessário para conexão já existe, mas por padrão está desabilitado. Para habilitá-lo, deve-se ir em “configurações”, selecionar “opções do desenvolvimento” e “Depuração de USB”.

Realizadas essas configurações, o dispositivo pode ser conectado por meio do ADB. A estrutura dos diretórios do sistema de arquivos do Android assemelha-se ao do Linux. A maioria dos dados de interesse forense estarão armazenadas no diretório /data/data.

Pelo método do ADB existem basicamente duas formas de se extrair os dados do Android. Usando o comando ADB push, que copiará os arquivos ativos (não apagados)

para o computador do especialista ou uma forma mais interessante, que é fazer uma cópia física dos dados.

Para realizar a cópia física, deve-se usar algum programa que possibilite esse tipo de cópia e enviar o resultado para o computador do perito. Uma opção é usar os programas dd e netcat, o primeiro para copiar os dados, o segundo para transmiti-los via uma conexão entre o dispositivo móvel e o computador. Esses dois programas não estão presentes na configuração padrão do Android, mas podem ser encontrados na Internet e baixados para o dispositivo por meio dos comandos ADB.

Os pontos fracos dessa abordagem de extração de dados são a necessidade de habilitar o modo de depuração de USB no próprio dispositivo e a necessidade de se ter acesso de super-usuário (*root*). Caso o dispositivo esteja protegido por senha, não será possível habilitar a depuração USB. Dessa forma, um outro meio de extração de dados é necessário.

2.8.1.3. Acessando dispositivo Android por meio de substituição de partição de recuperação (*recovery partition*)

Os dispositivos Android podem ser inicializados em outros modos além do modo padrão. Um modo de inicialização importante é o modo *recovery*. A inicialização em um modo diverso do padrão pode ser realizada pressionando uma combinação de teclas, que varia de fabricante para fabricante, durante o processo de *boot*.

O software de fábrica que está instalado na partição *recovery*, também conhecido como *stock recovery*, permite que o dispositivo seja restaurado para a configuração de fábrica, no qual todos os dados do usuário serão apagados, fazer atualizações do sistema, entre outros. Não há nenhuma opção na *stock recovery* que seja de interesse para a área forense.

Porém, a *stock recovery* pode ser substituída por um programa alternativo, geralmente chamado de *custom recovery*. Por meio da *custom recovery* é possível fazer o *backup* dos dados, ganhar acesso de *root* e outras atividades de interesse pericial. As principais *custom recoveries* encontradas na Internet atualmente são a CWM (ClockWorkMod) e TWRP (TeamWin, https://twrp.me/). Cada modelo de dispositivo móvel necessitará de uma *custom recovery* apropriada. Uma fonte de informação sobre as *custom recoveries* é http://forum.xda-developers.com/.

Um comparativo das funcionalidades disponíveis nesse dois tipos de *recoveries* é apresentado nas figuras 2.23 e 2.24.

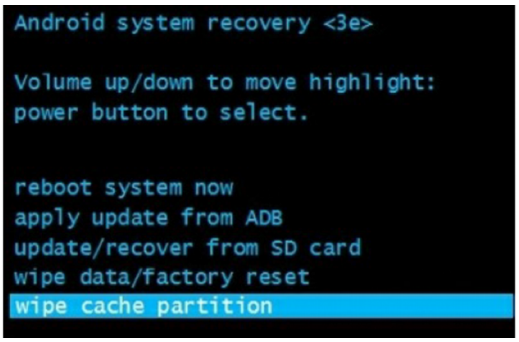


Figura 2.24. ***Stock recovery***

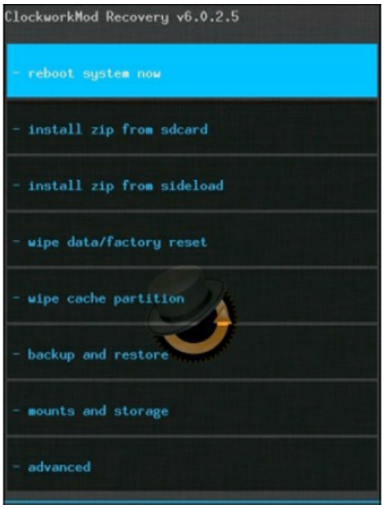


Figura 2.23: ***Custom recovery***

Referências

Brooks, Charles L. - CHFI Computer Hacking Forensic Investigator Certification All-in-One Exam Guide; 2014

Carrier, Brian – File System Analysis – USA, 2006

Carvey, Harlan – Windows Forensic Analysis – USA: Syngress, 2009

Epifani, Mattia; Stirparo, Pasquale - Learning iOS Forensics – Packet Publishing: 2015

Gomes, Jeremias Moreira – A forense computacional e os discos de estado sólido – ICoFCS, 2012

Fagundes, Leonardo L.; Neukamp, Paulo A.; da Silva, Pamela C. – Ensino da Forense Digital Baseado em Ferramentas Open Source – ICoFCS, 2011

Machado, Margarida Helena Serejo. A Regulamentação da Cadeia de Custódia na Ação Penal: Uma necessidade Premente. Corpo Delito, n.1, p. 18-23, Brasília, 2009.

Marimoto, Carlos Eduardo - Hardware II – O Guia Definitivo – Sul Editores - Porto Alegre, 2010

Merola, Antonio - Data Carving Concepts - SANS Institute – November, 2008

Silva, Gilson Marques; Lorens, Evandro Mário – Extração e Análise de Dados em Memória na Perícia Forense Computacional – ICoFCS, 2009

Stallings, Willian – Criptografia e segurança de redes – 4ª Edição – São Paulo, 2008

Tamma , Rohit; Tindall, Donnie – Learning Android Forensics – Packet Publishing: 2015

Velho, Jesus Antonio; et al – Tratado de Computação Forense – Millenium Editora; São Paulo, 2016

Yiannis, Chrysanthou - Modern Password Cracking: A hands-on approach to creating an optimised and versatile attack. Technical Report RHUL–MA–2013– 7; 2013

Capítulo 3

Canais laterais em criptografia simétrica e de curvas elípticas: ataques e contramedidas

Lucas Z. Ladeira, Erick N. Nascimento, João Paulo F. Ventura,
Ricardo Dahab, Diego F. Aranha, Julio C. López Hernández

Resumo

Para prover segurança em um ambiente hostil, algoritmos criptográficos precisam resistir a uma infinidade de ataques buscando a obtenção de informação sigilosa, acesso não-autorizado, entre outros. Tais ataques ocorrem tanto nos algoritmos e seus problemas computacionais subjacentes, quanto nas implementações de um criptossistema. Uma classe de ataques sobre implementações de algoritmos são os chamados ataques de canal lateral, que fazem uso de informações vazadas durante a execução de uma primitiva criptográfica. Ataques dessa natureza utilizam variações no tempo de execução, no consumo de energia, emanações eletromagnéticas e outras características do dispositivo alvo. Contramedidas para esses ataques podem ser baseadas em modificações no software ou no hardware. Neste minicurso, discutem-se ataques e contramedidas em implementações em software de métodos criptográficos simétricos, e assimétricos baseados em curvas elípticas.

Abstract

In order to provide security in a hostile environment, cryptographic algorithms must resist many attacks aiming to capture confidential information, obtain non-authorized access, among others. These attacks can target either the algorithms and their underlying hardness assumptions, or the implementations of a cryptosystem. Side-channel attacks are a class of attacks directed at the implementations of cryptographic algorithms and make use of information leaked during the execution of a cryptographic primitive. Such attacks are based on variations in execution time, energy consumption, electromagnetic emanations and other features of the device. Countermeasures are based on software or hardware mechanisms, or both. In this short course, we discuss side-channel attacks and countermeasures against software implementations of symmetric primitives and curve-based public-key cryptography.

3.1. Introdução

A Internet das Coisas (IoT, do inglês *Internet of Things*) permite a interconexão de múltiplos dispositivos embarcados que manipulam dados de diferentes tipos e realizam tarefas diversas, algumas até críticas. Este ambiente promete trazer enormes benefícios para a vida em sociedade e, naturalmente, induz novos requisitos para sua implementação eficaz e robusta. Entre estes requisitos, segurança, tolerância a falhas e privacidade surgem como dimensões novas e fundamentais no projeto de sistemas embarcados.

Apesar da importância de se equipar a Internet das Coisas com mecanismos robustos de segurança, a mudança de paradigma trazida por essa nova tecnologia cria um problema desafiador para o projeto de mecanismos de segurança: enquanto os dispositivos precisam permanecer compactos e baratos, a quantidade massiva de dados coletados e transportados por esses dispositivos e sua natureza sensível certamente terão implicações significativas em privacidade. Simultaneamente, qualquer solução prática precisa levar em conta os recursos reduzidos e a proteção física limitada que são típicas de dispositivos da Internet das Coisas.

Este minicurso discute técnicas para implementação segura de algoritmos criptográficos, com aplicações para a proteção de dispositivos embarcados operando na Internet das Coisas. Pretende-se discutir duas classes de algoritmos, criptografia simétrica baseada em cifras de bloco e funções de resumo criptográfico e criptografia assimétrica baseada em curvas elípticas, abrangendo seu projeto e implementação eficiente e segura contra ataques de canal lateral. A motivação para o estudo de primitivas simétricas é o menor consumo de recursos que seus correspondentes assimétricos, resultando em maior eficiência e tempo de vida quando sua utilização é maximizada em protocolos de comunicação para a Internet das Coisas. Por outro lado, esquemas para criptografia assimétrica são essenciais para estabelecer chaves criptográficas para primitivas simétricas e, portanto, precisam ser parte integral para qualquer arquitetura aplicada de segurança. Nesse cenário, o estudo de esquemas assimétricos baseados em curvas elípticas são de interesse especial, devido ao seu potencial para implementação em dispositivos com recursos restritos.

Sobre o conteúdo este documento

Este minicurso é uma versão atualizada e aprofundada de um minicurso apresentado no SBSeg 2009 por dois dos autores deste texto. Maior ênfase é dada a experimentos reais e contramedidas aos ataques de canais laterais, bem como aos métodos criptográficos simétricos. Quanto aos assimétricos, a maior parte é devotada aos métodos baseados em curvas elípticas sobre corpos primos, e tão somente à aritmética de pontos da curva elíptica, sem nos atermos à aritmética de corpos finitos.

Um bom número de termos técnicos foram mantidos em inglês, em razão de não termos um vocabulário estável da área em português, ainda que vários termos tenham já boas traduções. Procuramos manter os termos em inglês em fonte itálico.

Agradecimentos

Os autores gostariam de agradecer: ao comitê de programa dos minicursos do SBSeg 2016 pela oportunidade de apresentarmos este trabalho; à Fapesp e Intel pelo suporte financeiro ao projeto "Secure execution of cryptographic algorithms"; à Intel pelo generoso apoio ao projeto de pesquisa "Software Implementation of Cryptographic Algorithms"; ao CNPq/Intel pelas bolsas concedidas no contexto deste último projeto.

3.2. Criptografia simétrica

Algoritmos criptográficos podem ser classificados em *simétricos* e *assimétricos*. Os algoritmos simétricos baseiam-se na existência de um segredo pré-compartilhado, chamado de *chave secreta*. Em um contexto de sigilo, chaves de encriptação e decriptação são idênticas ou podem ser calculadas eficientemente uma a partir da outra. O tarefa (difícil) de um adversário nesses esquemas normalmente resume-se a explorar o espaço de chaves, de tamanho gigantesco, em um ataque chamado de *busca exaustiva*. O algoritmo AES (*Advanced Encryption Standard*) [AES 2001] é talvez o mais difundido entre os algoritmos simétricos de encriptação em uso, enquanto SHA [SHA 2012] é uma família padronizada de funções de resumo criptográfico. Algoritmos assimétricos empregam um par de chaves, *pública* e *privada*, que são calculadas de forma a existir uma relação especial entre ambas: calcular a chave pública a partir da chave privada é eficiente, mas não se conhece algoritmo eficiente para resolver o problema no sentido inverso. Assim, a chave pública pode ser utilizada para encriptação ou verificação de assinaturas, sem ameaça à segurança do algoritmo criptográfico, e a chave privada para decriptação ou assinatura digital deve ser mantida em sigilo, sob posse exclusiva do seu detentor. Criptografia de curvas elípticas (ECC – *Elliptic Curve Cryptography*) é um exemplo de família de algoritmos assimétricos.

3.2.1. Encriptação

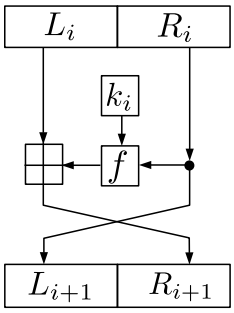
O requisito de segurança de *confidencialidade* pode ser provido por um par de funções de encriptação E e decriptação D, ambas parametrizadas por uma chave k. A encriptação de uma mensagem M sob a chave k produz um criptograma $C = E_k(M)$; assim, a mensagem original pode ser recuperada pela função de decriptação como $M = D_k(C)$, de forma que a propriedade de consistência seja mantida, isto é, $M = D_k(E_k(M))$ para quaisquer valores de M e k.

As funções E e D podem ser implementadas por um algoritmo simétrico de encriptação de duas formas distintas. *Cifras de bloco*, como o AES, especificam como encriptar uma sequência de *bits* com tamanho fixo, e precisam ser estendidos para mensagens de tamanho arbitrário por meio de um *modo de operação*. *Cifras de fluxo*, como ChaCha [Bernstein 2005], calculam uma cadeia de chaves a partir da chave k, que é então combinada com a mensagem original utilizando operações `XOR` (OU exclusivo). É desejável que criptogramas transmitidos em canais de comunicação sejam acompanhados de *autenticadores*, que permitem ao destinatário certificar-se de que o remetente é a origem legítima da mensagem e verificar sua integridade.

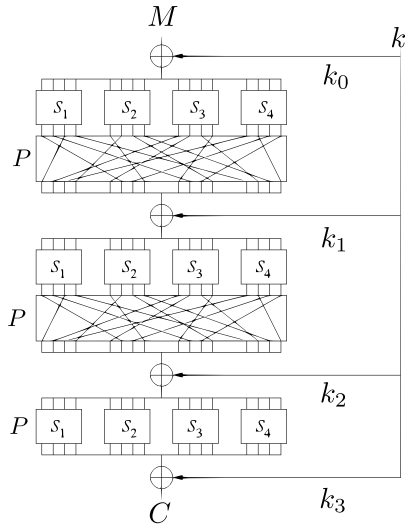
Há muitos paradigmas diferentes para se construir cifras simétricas. Em geral,

os algoritmos seguem a idéia proposta por Shannon [Shannon 1949] de um *produto iterado de cifras*, em que uma *função de rodada* composta por cifras menores e parametrizada por *chaves de rodada* é repetida um número fixo r de vezes. A função de rodada combina pequenas *cifras de substituição*, que substituem símbolos para confundir a relação entre o criptograma e a chave criptográfica, com uma *cifra de transposição*, que altera a ordem dos *bits* para espalhar a redundância do texto claro ao longo do criptograma [Shannon 1949]. O cálculo de chaves de rodada k_i a partir da chave k é também chamado de *escalonamento de chaves*.

Os paradigmas clássicos mais comuns para construção de cifras de bloco são *Redes de Feistel* e *Redes de Substituição-Permutação* (SPN – *Substitution-Permutation Network*). Redes de Feistel (ou cifras de Luby-Rackoff [Luby and Rackoff 1988]) utilizam uma função de rodada não-inversível f que atua sobre o bloco de texto claro $(L_0 \parallel R_0)$. A cada rodada, a função f calcula a próxima metade do estado interno pela regra $R_{i+1} = L_i \oplus f(R_i, k_i)$, enquanto a metade restante é uma simples cópia $L_{i+1} = R_i$, até que seja atingido o bloco de criptograma $(L_r \parallel R_r)$. A decriptação segue processo análogo, utilizando as regras inversas $R_i = L_{i+1}$ e $L_{i+1} = R_{i+1} \oplus f(L_{i+1}, k_i)$. SPNs alternam *caixas de substituição* S_i que operam sobre pedaços de ℓ *bits* do texto claro (que implementam cifras de substituição) com uma *camada de difusão linear* P que opera sobre todo o estado interno. Ao final de cada rodada, a chave de rodada k_i é adicionada ao estado interno por uma operação `XOR`. A decriptação é realizada pela aplicação de operações inversas também na ordem inversa. As Figuras 3.1a e 3.1b apresentam diagramas para Redes de Feistel e de Permutação-Substituição, respectivamente.



(a) Rodada de uma Rede de Feistel.



(b) Rede de Permutação-Substituição.

Figura 3.1: Construções para cifras de bloco.

Cifras de fluxo, por sua vez, possuem construções derivadas de geradores pseudo-aleatórios, onde a chave criptográfica faz o papel da semente, a qual é expandida numa sequência pseudo-aleatória de bits, formando uma chave muito mais longa. Estas devem ser construídas de modo a maximizar o período do gerador, para evitar repetições na chave expandida. Pode-se dizer que cifras de fluxo são um relaxamento do *one-time pad* (OTP),

onde cada símbolo do texto claro é combinado com um símbolo aleatório da chave, que deve possuir o mesmo tamanho da mensagem a ser encriptada. A chave pseudo-aleatória de uma cifra de fluxo abre mão da *segurança incondicional* do OTP [Shannon 1949], em favor de uma premissa de segurança computacional com ganho de eficiência.

Modos de operação

Modos de operação estendem o requisito de confidencialidade para mensagens de tamanho arbitrário, ou até mesmo requisitos de segurança adicionais, como *encriptação autenticada*, em um único passo. Estes modos dividem a mensagem M em n blocos de mesmo tamanho $\{M_1, \ldots, M_n\}$ e utilizam uma regra para produzir os blocos de criptograma $C = \{C_1, \ldots, C_n\}$. O último bloco precisa ser completado com *preenchimento* (ou *padding*) para poder ser processado corretamente. Diversos modos de operação já foram padronizados por agências de padronização como o NIST para utilização governamental ou na indústria, dentre eles CBC, CTR e GCM.

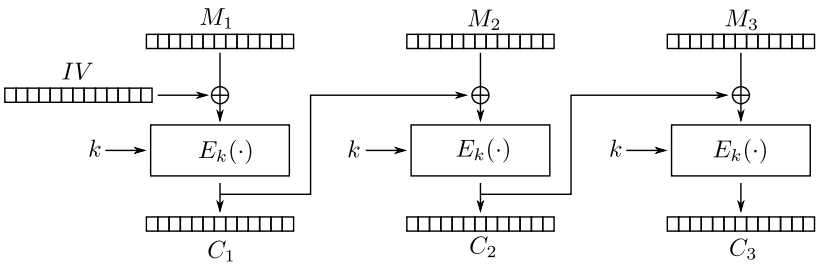
No modo de operação CBC (*Cipher Block Chaining*), o próximo bloco de criptograma é calculado a partir de um bloco de texto claro e o bloco de criptograma anterior, pela regra $C_i = E_k(M_i \oplus C_{i-1})$, para $2 \leq i \leq n$. O bloco C_0 é definido como um *vetor de inicialização IV* único e imprevisível, que não deve se repetir em encriptações distintas. Observe que o encadeamento da encriptação aleatoriza o bloco de texto claro antes da próxima encriptação, fazendo com que blocos idênticos produzam blocos distintos no criptograma. Como apenas a decriptação de blocos distintos pode ser feita de modo independente, paralelismo pode ser extraído apenas no processo de decriptação.

O modo de operação CTR (*Counter* simula uma cifra de fluxo a partir de uma cifra de bloco. Os blocos do criptograma são calculados pela operação `XOR` entre blocos de texto claro e encriptação de valores consecutivos do vetor de inicialização IV. Portanto, a regra é dada por $C_i = M_i \oplus E_k(IV + i)$, para $1 \leq i \leq n$. Como o processamento de cada bloco é independente, este modo de operação oferece paralelismo tanto na encriptação quanto decriptação, sendo tipicamente mais eficiente que o modo CBC. A Figura 3.2 ilustra os dois modos de operação.

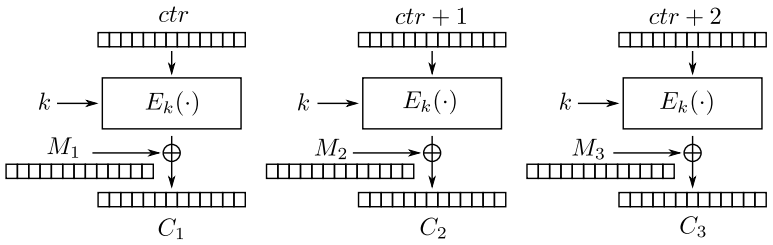
O modo de encriptação autenticada *GCM* (*Galois/Counter Mode*) [McGrew and Viega 2004] utiliza o modo CTR para encriptação e atualiza um autenticador calculado a partir dos blocos de criptograma C_i que permite detetar posteriormente qualquer manipulação do criptograma em trânsito, com alta probabilidade. Opcionalmente, o modo também autentica dados associados transmitidos às claras, funcionalidade útil para autenticar cabeçalhos de pacotes de rede ou outra informação pública cuja integridade deva ser preservada e possa ser verificada pelo destinatário.

3.2.2. Funções de *hash* ou resumo criptográfico

O objetivo de funções de *hash* ou resumo criptográfico é mapear uma cadeia de *bits* de tamanho arbitrário em um *resumo* com tamanho fixo em *bits*. Este resumo tem como papel identificar unicamente a mensagem de entrada, servindo como uma espécie de "impressão digital" da mesma. São muitas as aplicações em Criptografia e Segurança Computacional de funções criptográficas: armazenamento seguro de senhas, esquemas de assinatura di-



(a) Encriptação no modo CBC.



(b) Encriptação no modo CTR.

Figura 3.2: Modos de operação para cifras de bloco.

gital, verificação de integridade, derivação de chaves criptográficas, geração de números pseudo-aleatórios, projeto de cifras de fluxo e autenticadores, provas de trabalho de moedas criptográficas, entre outras. A Figura 3.3 apresenta essa funcionalidade abstrata de funções de resumo criptográfico.

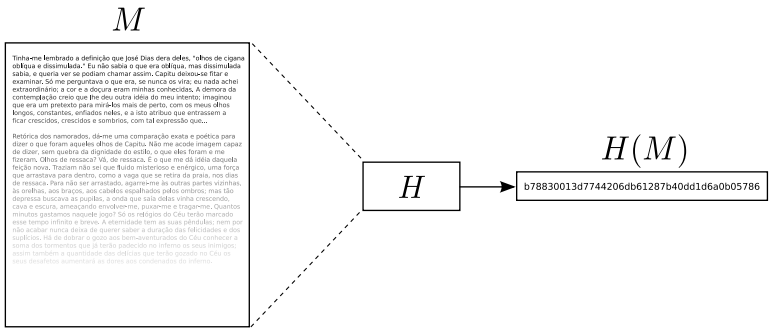


Figura 3.3: Função de resumo criptográfico mapeando M para o valor de resumo $H(M)$.

Em termos matemáticos, funções de resumo são da forma $H : \{0,1\}^* \to \{0,1\}^n$. Ou seja, mapeiam uma pré-imagem x de tamanho finito e arbitrário em um *valor de hash* $y = H(x)$ de tamanho fixo. A princípio, tratam-se de funções não-chaveadas, ou seja, não-parametrizadas por uma chave criptográfica, mas é claro que podem ser utilizadas também para calcular resumos de informação secreta. Para haver interesse criptográfico,

funções de resumo precisam satisfazer as três propriedades abaixo:

- **Resistência à pré-imagem**: Dada uma função de resumo criptográfico H e um resumo y, deve ser computacionalmente inviável encontrar x tal que $y = H(x)$. Em outras palavras, deve ser difícil "inverter" a função para um certo valor de resumo.
- **Resistência à segunda pré-imagem**: Dado um resumo y e uma mensagem x tais que $y = H(x)$, deve ser computacionalmente inviável encontrar uma mensagem $x' \neq x$ tal que $H(x') = H(x) = y$. Ou seja, deve ser difícil encontrar uma segunda mensagem mapeada para o mesmo valor de resumo.
- **Resistência a colisões**: Deve ser computacionalmente inviável encontrar mensagens x, x' tais que $H(x) = H(x')$; ou seja, deve ser difícil encontrar duas mensagens que colidem sob H para um mesmo valor de resumo.

Há muitas formas diferentes de se construir essas funções. Observe que, pelo tamanho e natureza dos conjuntos de domínio e imagem, *colisões* entre diferentes entradas sob uma função h necessariamente irão existir. O desafio é projetar uma função que torne computacionalmente inviável encontrar essas colisões, mesmo que seu número seja infinito. Classicamente, o paradigma Merkle-Damgård constrói uma função de resumo criptográfico resistente a colisão a partir de uma função de compressão h com entrada de tamanho fixo, também resistente a colisão [Merkle 1979, Damgård 1989]. Na Figura 3.4, a entrada X é dividida entre B blocos, com aplicação de preenchimento para formar o último bloco, e a saída da última função de compressão define o valor de resumo. Funções de resumo podem também ser construídas a partir de cifras de bloco [Preneel et al. 1993], ou problemas difíceis de Teoria dos Números, como a fatoração de inteiros.

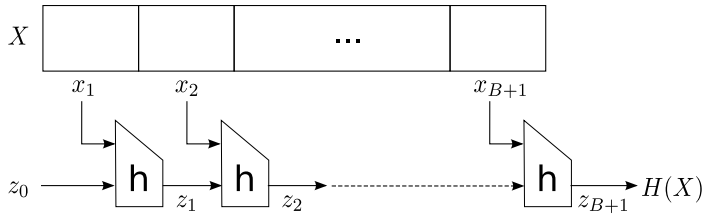


Figura 3.4: Construção Merkle-Damgård para funções de resumo criptográfico.

Um paradigma de construção que se tornou imensamente popular é a classe de *esponjas criptográficas*. Uma esponja é uma função com estado interno s de tamanho finito b, dividido em duas seções: a taxa de *bits* de tamanho r e a capacidade de tamanho c, tais que $b = c + r$. Completam a especificação uma função de permutação f de tamanho fixo, que transforma o estado interno, e uma regra de preenchimento. Durante a inicialização, uma função esponja atribui o valor 0 ao estado interno e completa a mensagem de entrada para que seu tamanho seja múltiplo de r, de forma que a entrada possa ser dividida em blocos de tamanho r. A cada iteração da função, um novo bloco da entrada é adicionado a parte do estado interno pela operação `XOR` e o estado interno s é substituído por $f(s)$. Em outras palavras, os blocos da entrada são absorvidos sucessivamente no estado interno. Na etapa final, toma-se r *bits* do estado interno por iteração até que a saída atinja

um tamanho pré-determinado, alternando com novas aplicações da função de permutação $f(s)$. A resistência a colisões ou ataques de pré-imagem depende do tamanho c, tipicamente escolhido como duas vezes o nível de segurança desejado. A Figura 3.5 apresenta a construção [Bertoni et al. 2008].

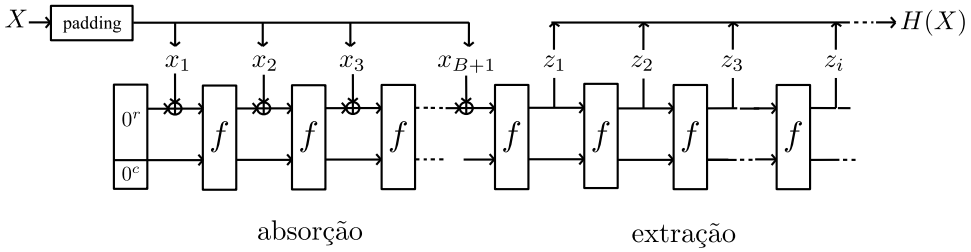
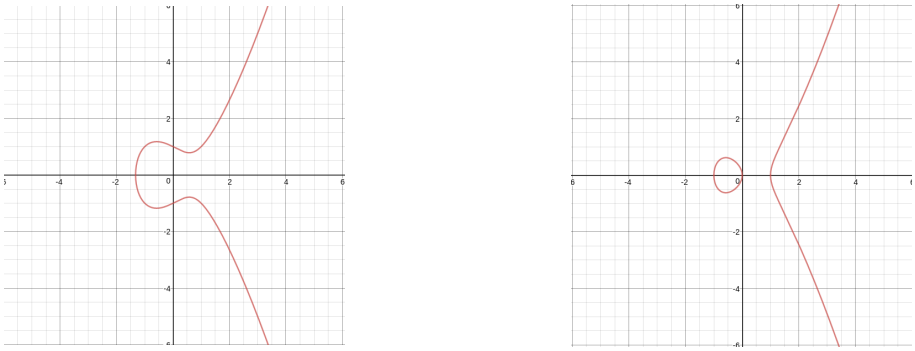


Figura 3.5: Esponja criptográfica, e suas fases de absorção e extração.

3.3. Criptografia de curvas elípticas (ECC)

Em 1985, Neal Koblitz [Koblitz 1987] e Victor Miller [Miller 1986], de forma independente, propuseram a utilização de curvas elípticas para projetar criptossistemas de chave-pública.

Curvas elípticas são definidas por equações cúbicas, como por exemplo $y^2 = x^3 - x$, onde os valores x, y pertencem a uma estrutura algébrica chamada corpo, dos quais os números racionais, reais e complexos são exemplos. A Figura 3.6 ilustra duas curvas elipticas sobre o conjunto $\mathbb{R}$ dos reais. Curvas elípticas não são importantes apenas para a Criptografia, mas também em outros ramos da Matemática como, por exemplo, na prova do Último Teorema de Fermat. [Hankerson et al. 2003]. No caso da Criptografia



(a) $y^2 = x^3 - x + 1$ (b) $y^2 = x^3 - x$

Figura 3.6: Exemplo de duas curvas elípticas

de Curvas Elípticas (ECC), o corpo sobre o qual curvas são definidas são os chamados corpos finitos. Tais corpos são conjuntos finitos de tamanho p^m, denotados $\mathbb{F}_{p^m}$, onde p um número primo e m um inteiro maior ou igual a 1, e munidos de duas operações $+, \times$. Os elementos de $\mathbb{F}_{p^m}$ são os polinômios de grau máximo $m-1$ e coeficientes em $Z_p = \{0, 1, 2, \ldots, p-1\}$. As operações $+, \times$ são as operações usuais de adição e

multiplicação de polinômios, com duas condições extras: (i) os coeficientes do polinômio $r(x)$ resultante de uma operação são reduzidos módulo p; e (ii) $r(x)$ é tomado módulo um polinômio irredutível $f(x)$ em Z_p.

Quando $m = 1$, dizemos $\mathbb{F}_p$ é um corpo primo e seus elementos são simplesmente os inteiros em $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$ com aritmética módulo p. A seguir apresentaremos equações de curvas elípticas para corpos primos $\mathbb{Z}_p$ e *corpos binários* $\mathbb{F}_{2^m}$.

Curvas sobre $\mathbb{F}_p$ e $\mathbb{F}_{2^m}$

Uma *curva elíptica* E sobre $\mathbb{F}_p$ é definida por uma equação da forma

$$y^2 = x^3 + ax + b, \tag{1}$$

onde os coeficientes $a, b \in \mathbb{F}_p$ satisfazem $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$. Para corpos binários $\mathbb{F}_{2^m}$, uma curva elíptica (não-supersingular) é definida por uma equação da forma

$$y^2 + xy = x^3 + ax^2 + b, \tag{2}$$

onde os coeficientes $a, b \in \mathbb{F}_{2^m}$ e $b \neq 0$. Dizemos que um par (x, y), onde $x, y \in \mathbb{F}_q$ ($\mathbb{F}_p$ ou $\mathbb{F}_{2^m}$), é um *ponto* na curva se (x, y) satisfaz a equação 1 (ou 2). O conjunto de pontos da curva junto com um ponto especial ∞, chamado *ponto no infinito*, é denotado por $E(\mathbb{F}_q)$. Por exemplo, se E é a curva elíptica sobre $\mathbb{Z}_{29}$, definida pela equação

$$y^2 = x^3 + 4x + 20, \tag{3}$$

então o conjunto de 37 pontos de $E(\mathbb{F}_{29})$ é

$$\begin{array}{lllllll}
\{\infty, & (2,6), & (4,19), & (8,10), & (13,23), & (16,2), & (19,16), \\
(0,7), & (2,23), & (5,7), & (8,19), & (14,6), & (16,27), & (20,3), \\
(0,22), & (3,1), & (5,22), & (10,4), & (14,23), & (17,10), & (20,26), \\
(1,5), & (3,28), & (6,12), & (10,25), & (15,2), & (17,19), & (24,7), \\
(1,24), & (4,10), & (6,17), & (13,6), & (15,27), & (19,13), & (24,22), \\
(27,2), & (27,27)\}. & & & & &
\end{array}$$

Existe uma operação, conhecida como lei de "secantes e tangentes", que permite "somar" pontos elípticos em E. Com essa operação, o conjunto de pontos de $E(\mathbb{F}_q)$ forma um grupo abeliano cuja identidade é o ponto no infinito ∞. A fórmula para somar pontos requer umas poucas operações aritméticas no corpo finito subjacente. Para corpos primos $\mathbb{F}_p$, as fórmulas explícitas para somar dois pontos são dadas a seguir:

1. *Identidade:* $P + \infty = \infty + P = P$ para todo $P \in E(\mathbb{F}_p)$
2. *Negativos:* Se $P = (x, y) \in E(\mathbb{F}_p)$ então $(x, y) + (x, -y) = \infty$.
   O ponto $(x, -y)$ é denotado por $-P$ e chamado o *negativo* de P.
3. *Soma de pontos:* Seja $P = (x_1, y_1) \in E(\mathbb{F}_p)$ e $Q = (x_2, y_2) \in E(\mathbb{F}_p)$, onde $P \neq \pm Q$. Então $P + Q = (x_3, y_3)$, onde

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \quad \text{e} \quad y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1.$$

4. *Duplicação de ponto:* Seja $P = (x_1, y_1) \in E(\mathbb{F}_p)$, onde $P \neq -P$. Então $2P = P + P = (x_3, y_3)$, onde

$$x_3 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1 \text{ e } y_3 = (\frac{3x_1^2 + a}{2y_1})(x_1 - x_3) - y_1.$$

A partir da operação de soma no grupo $E(\mathbb{F}_q)$, define-se o produto de um escalar $k \in \mathbb{Z}$ por um ponto $P \in E(\mathbb{F}_q)$ como o ponto $kP = P + P + \cdots + P$ (com k parcelas) se $k > 0$, e por extensão $0P = \infty$ e $-kP = k(-P) = -(kP)$. Esta operação é chamada de *multiplicação de um ponto por um escalar* e é a operação central dos esquemas criptográficos baseados em curvas elípticas. Quando estiver subentendido o corpo sobre o qual a curva $E(\mathbb{F}_{p^m})$ é definida, escreveremos simplesmente E para denotar a curva.

3.3.1. Curvas NIST e curvas modernas

As curvas NIST foram geradas pela NSA (National Security Agency) dos EUA e recomendadas para utilização pelo governo americano. São dez escolhas para corpos finitos, sendo cinco corpos primos ($\mathbb{F}_p$) e cinco corpos binários ($\mathbb{F}_{2^m}$) [Brown et al. 2001]. Os corpos finitos recomendados são relacionados a seguir, com os respectivos polinômios irredutíveis, onde o prefixo P denota corpos primos e o B denota corpos binários:

- P-192
  $\mathbb{F}_{192}$ $p = 2^{192} - 2^{64} - 1$;
- P-224
  $\mathbb{F}_{224}$ $p = 2^{224} - 2^{96} + 1$;
- P-256
  $\mathbb{F}_{256}$ $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$;
- P-384
  $\mathbb{F}_{384}$ $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$;
- P-521
  $\mathbb{F}_{521}$ $p = 2^{521} - 1$;
- B-163
  $\mathbb{F}_{2^{163}}$ $f(x) = x^{163} + x^7 + x^6 + x^3 + 1$;
- B-233
  $\mathbb{F}_{2^{233}}$ $f(x) = x^{233} + x^{74} + 1$;
- B-283
  $\mathbb{F}_{2^{283}}$ $f(x) = x^{283} + x^{12} + x^7 + x^5 + 1$;
- B-409
  $\mathbb{F}_{2^{409}}$ $f(x) = x^{409} + x^{87} + 1$;
- B-571
  $\mathbb{F}_{2^{571}}$ $f(x) = x^{571} + x^{10} + x^5 + x^2 + 1$.

Os corpos foram determinados com base no desempenho, facilitando a aritmética utilizada. Há também opções de curvas binárias genéricas (ou de Koblitz) para as curvas binárias. As curvas acima foram geradas de forma pseudo-aleatória, descartando-se curvas que não fossem resistentes aos ataques conhecidos. Nem todas as etapas do processo de geração das curvas são completamente transparentes, especialmente na escolha de sementes do gerador de bits pseudo-aleatórios. Essa lacuna tem produzido resistência na comunidade técnica, por haver a possibilidade de que as sementes tenham sido escolhidas de forma a produzir curvas com vulnerabilidades conhecidas pela NSA mas desconhecidas do público em geral.

Por esse motivo, outros modelos de curvas elípticas têm recebido maior atenção na literatura científica e na adoção pela indústria, por terem métodos de geração mais transparentes e implementações de alto desempenho e protegidas contra ataques de canal

lateral de tempo. Exemplos dessas são as curvas de Montgomery e Edward. Nessas curvas, o número de operações realizadas é *regular* e baseiam-se apenas no comprimento das chaves e não no valor dos *bits* da chave.

A curva de Montgomery é dada pela equação $E : y^2 = x^3 + Ax^2 + x$. O valor de A pode ser alterado de forma a melhorar o desempenho das multiplicações escalares. O corpo primo mais utilizado com essa curva é $\mathbb{F}_{2^{255}-19}$ com o valor de $A = 486662$ [Düll et al. 2015]. Essa combinação possibilita a implementação eficiente da multiplicação por escalar usando o método de *Montgomery Ladder*.

A curva de Edwards é encontrada na literatura na sua forma *twisted*, em razão do desempenho exibido na combinação com os primos de Mersenne $p = 2^{127} - 1$ e 2^{255-10} em [Costello and Longa 2015] e [Bernstein et al. 2012], respectivamente. A equação dessa curva é $-x^2 + y^2 = 1 + dx^2y^2$, onde d é um não-quadrado em $\mathbb{F}_p^2$.

Nessa curva, a soma de dois pontos segue a fórmula de adição de *Edwards*:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + x_2y_1}{1 + xd_1x_2y_1y_2}, \frac{y_1y_2 + x_1x_2}{1 - dx_1x_2y_1y_2}\right)$$

Um fato interessante é que a duplicação de um ponto tem a mesma fórmula da adição, o que não acontece em geral.

3.3.2. Algoritmos básicos para multiplicação por escalar (ECSM)

Como já dissemos, a principal operação em curvas elípticas é a multiplicação por escalar kP. O escalar k é, de fato, a chave privada, ou parte dela, em vários esquemas criptográficos baseados em ECC. Dessa forma, é necessário explorar diferentes formas de implementar esse cálculo, não só do ponto de vista de desempenho como o de segurança, assunto central deste texto. Nesta seção serão apresentados os métodos mais simples para o cálculo de kP, resistentes ou não a ataques de canal lateral, e pequenas variações.

O primeiro é o método *Double-and-add* (também conhecido como *Double-and-add-not-always*) [Rivain 2011]. O Algoritmo 1 é chamado de *left-to-right, double-and-add* já que os bits de k são processados da esquerda para a direita. Esse algoritmo não é resistente a ataques de canal lateral. Uma simples inspeção do código revela que o comando na linha 5 terá tempos de execução diferentes dependendo do valor do bit k_i.

Algoritmo 1 Left-to-right double-and-add

Entrada: $P \in E(\mathbb{F}_p), k = (k_{t-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$
1: $N \leftarrow P$
2: $Q \leftarrow 0$
3: **for** i from $t-1$ **downto** 0 **do**
4: $\quad N \leftarrow 2N$
5: $\quad$ **if** $k_i = 1$ **then**
6: $\quad\quad Q \leftarrow Q + N$
7: $\quad$ **end if**
8: **end for**
9: **return** Q

Uma outra maneira de percorrer os *bits* de k é da direita para a esquerda, conhecida como *right-to-left* [Rivain 2011], resultando num algoritmo ligeiramente diferente, o Algoritmo 2. Pelas mesmas razões acima, esse algoritmo não resiste a ataques de canal lateral de tempo.

Algoritmo 2 Right-to-left double-and-add

Entrada: $P \in E(\mathbb{F}_p), k = (k_{t-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$
1: $N \leftarrow P$
2: $Q \leftarrow 0$
3: **for** i from 0 **to** $t-1$ **do**
4: **if** $k_i = 1$ **then**
5: $Q \leftarrow Q + N$
6: **end if**
7: $N \leftarrow 2N$
8: **end for**
9: **return** Q

Mais adiante trataremos de outros métodos de multiplicação por escalar que são resistentes a ataques de canal lateral, tornando constante o número de operações executadas em cada iteração e o tempo de execução independente do valor do escalar k.

3.3.3. Algoritmos para multiplicação de ponto fixo por escalar

Nos casos em que P é um ponto fixo, isto é, reutilizado para diferentes valores de k, é possível pré-processar parte da multiplicação escalar kP, obtendo um ganho de desempenho [Hankerson et al. 2003].

A ideia é pré-computar os valores $2^wP, 2^{2w}P, 2^{3w}P, \ldots, 2^{w(d-1)}P$, onde onde w é o tamanho da palavra do processador-alvo da implementação, usualmente 32 ou 64 bits, também chamada de janela ou *window* em inglês, e d é o número de tais palavras ocupadas pelo escalar k [Hankerson et al. 2003]; isto é, se $k = (k_{t-1}, \ldots, k_1, k_0)_2$ tem k bits, então $d = \lceil t/w \rceil$, e escrevemos $k = (K_{d-1}, \ldots, K_1, K_0)_{2^w}$ para denotar os blocos de w bits da chave k.

Utilizar janelas é uma prática comum para aumento da eficiência de implementações, já que operandos podem ser descritos e manipulados em blocos de bits do tamanho da janela, para o qual as instruções do processador já são especializadas. O Algoritmo 3, proposto por Brickell at al. utiliza essa técnica, conhecida por *Fixed-base Windowing Method* .

Algoritmo 3 *Fixed-base windowing method* para multiplicação escalar de um ponto

Entrada: Janela w, $P \in E(\mathbb{F}_p)$, $d = \lceil t/w \rceil$, $k = (K_{d-1}, \ldots, K_1, K_0)_{2^w} \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$

1: Compute $P_i = 2^i P, 0 \leq i \leq d-1$ (*pré-computação*)
2: $A \leftarrow \infty$, $B \leftarrow \infty$
3: **for** j from $2^w - 1$ **downto** 1 **do**
4: **for each** i for which $K_i = j$ **do**
5: $B \leftarrow B + P_i$
6: $A \leftarrow A + B$
7: **end for**
8: **end for**
9: **return** A

3.3.4. Algoritmos regulares para multiplicação por escalar

Para que algoritmos de multiplicação por escalar sejam resistentes a ataques de canal lateral é necessário que o fluxo de execução seja regular, isto é, que não haja variações de tempo em função do valor do escalar k, a chave privada.

O primeiro exemplo dessa regularidade é ilustrada no Algoritmo 4, chamado *Double-and-add-always* e proposto por Coron [Coron 1999], que executa uma adição e uma duplicação elípticas em todas a iterações. Uma implementação mais uniforme é a chamada *Atomic Double-and-Add* [Batina et al. 2014], que executa somente operações de soma de pontos elípticos.

Algoritmo 4 Double-and-add-always

Entrada: $P \in E(\mathbb{F}_p)$, $k = (k_{t-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$

1: $R_0 \leftarrow P$
2: **for** i from $t-2$ **downto** 0 **do**
3: $R_0 \leftarrow 2R_0$
4: $R_1 \leftarrow R_0 + P$
5: $b \leftarrow k_i$
6: $R_0 \leftarrow R_b$
7: **end for**
8: **return** R_0

Algoritmo 5 Atomic double-and-add

Entrada: $P \in E(\mathbb{F}_p)$, $k = (k_{t-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$
1: $R_0 \leftarrow P$
2: $R_1 \leftarrow P$
3: $j \leftarrow t-2$
4: $b \leftarrow 0$
5: **while** $j \geq 0$ **do**
6: $\quad R_0 \leftarrow R_0 + R_b$
7: $\quad b \leftarrow b \oplus k_j$
8: $\quad j \leftarrow j + k_j - 1$
9: **end while**
10: **return** R_0

Um outro algoritmo importante é o chamado *Montgomery Ladder* [Düll et al. 2015], que é mais eficiente que outros similares, e tem a vantagem adicional de poder ser otimizado para diferentes curvas. Primeiramente exibimos uma implementação genérica do método, no Algoritmo 6.

Algoritmo 6 *Montgomery Ladder*

Entrada: $P \in E(\mathbb{F}_p), k = (k_{t-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$
1: $R_0 \leftarrow P$
2: $R_1 \leftarrow 2P$
3: **for** $j \leftarrow t-2$ **to** 0 **do**
4: $\quad b \leftarrow k_j$
5: $\quad R_{1-b} \leftarrow R_0 + R_1$
6: $\quad R_b \leftarrow 2R_b$
7: **end for**
8: **return** R_0

Quando se usa o *Montgomery Ladder* com a curva de Montgomery com primo $p = 2^{255} - 19$, há um ganho de eficiência. Para o cálculo de kP, as entradas para o algoritmo são o escalar k com 256 bits, e a coordenada x_P do ponto P. A função *cswap*, ou *conditional swap*, faz a troca dos valores dos dois primeiros parâmetros caso o valor do terceiro parâmetro seja 1.

Algoritmo 7 *Montgomery ladder* particularizado para a curva de Montgomery

Entrada: Coordenada x_P de um ponto $P \in E(\mathbb{F}_p), k = (k_{255}, \ldots, k_1, k_0)_2 \in \mathbb{N}$.
Saída: (X_1, Z_1) tais que a coordenada x_{kP} de kP é igual a X_1/Z_1.

1: $X_1 \leftarrow 1$
2: $Z_1 \leftarrow 0$
3: $X_2 \leftarrow x_P$
4: $Z_2 \leftarrow 1$
5: $r \leftarrow 0$
6: **for** $i \leftarrow 254$ **to** 0 **do**
7: $\quad b \leftarrow k_i$
8: $\quad c \leftarrow b \oplus r$
9: $\quad r \leftarrow b$
10: $\quad (X_1, X_2) \leftarrow cswap(X_1, X_2, c)$
11: $\quad (Z_1, Z_2) \leftarrow cswap(Z_1, Z_2, c)$
12: $\quad (X_1, Z_1, X_2, Z_2) \leftarrow ladder(x_P, X_1, Z_1, X_2, Z_2)$
13: **end for**
14: **return** (X_1, Z_1)

A função *ladder*, detalhada no Algoritmo 8, calcula uma duplicação seguida por uma adição diferencial de pontos. Uma adição diferencial é uma adição de dois pontos conhecendo-se sua diferença. Não justificaremos esse fato aqui, nos limitando à sua descrição. A operação não utiliza valores pré-calculados pré-armazenados, dificultando ataques de tempo na latência da hierarquia de memória (cache). A quantia A na linha 10 é o coeficiente A da curva de Montgomery.

Algoritmo 8 Função *ladder* - duplicação e adição diferencial de pontos

Entrada: x, X_1, Z_1, X_2, Z_2
Saída: X_1, Z_1, X_2, Z_2

1: $T_1 \leftarrow X_2 + Z_2$
2: $X_2 \leftarrow X_2 - Z_2$
3: $Z_2 \leftarrow X_1 + Z_1$
4: $X_1 \leftarrow X_1 - Z_1$
5: $T_1 \leftarrow T_1 \cdot X_1$
6: $X_2 \leftarrow X_2 \cdot Z_2$
7: $Z_2 \leftarrow Z_2 \cdot Z_2$
8: $X_1 \leftarrow X_1 \cdot X_1$
9: $T_2 \leftarrow Z_2 - X_1$
10: $Z_1 \leftarrow T_2 \cdot (A+2)/4$
11: $Z_1 \leftarrow Z_1 + X_1$
12: $Z_1 \leftarrow T_2 \cdot Z_1$
13: $X_1 \leftarrow Z_2 \cdot X_1$
14: $Z_2 \leftarrow Z_2 - X_2$
15: $Z_2 \leftarrow Z_2 \cdot Z_2$
16: $Z_2 \leftarrow Z_2 \cdot x$
17: $X_2 \leftarrow T_1 + X_2$
18: $X_2 \leftarrow X_2 \cdot X_2$
19: **return** (X_1, Z_1, X_1, Z_2)

3.3.5. Protocolos de IoT e ECC

A necessidade por protocolos eficientes na Internet das Coisas é bem exemplificada no trabalho de Simplício et al. [Jr. et al. 2016], onde é descrita uma rede de sensores sem fio utilizando protocolos criptográficos baseados em ECC. Nós dessa rede têm baixa capacidade de recursos vitais como processamento, largura de banda, energia, memória, entre

outros. Além do mais, sendo nós sensores autônomos, há boa chance de que sejam fisicamente monitorados ou manipulados por adversários. Portanto, a comunicação entre os nós da rede requer não só implementações eficientes mas também robustas contra ataques por canais laterais, para que possam bem desempenhar suas funções em protocolos criptográficos para acordo de chaves, encriptação e decriptação, entre outras funções de segurança. Descreveremos a seguir alguns ataques por canais laterais.

3.4. Ataques de canal lateral

3.4.1. Ataques temporais

A premissa fundamental de ataques temporais é de que o tempo gasto na execução de uma instrução é influenciado por seus respectivos operandos [Hankerson et al. 2004]. Estudos mostraram [Brumley and Boneh 2003] a viabilidade desse ataque contra servidores executando protocolos como o SSL com RSA devido à latência da comunicação decorrente da rede local.

Como todos os ataques por canais laterais envolvem o monitoramento de uma grandeza física, um requisito para o sucesso de um ataque temporal é que as operações com a chave criptográfica sejam *lentas* o suficiente para serem medidas. Acreditava-se que esse tipo de ataque seria possível apenas em operações de rede ou rádio e jamais em processadores de dispositivos móveis ou em computadores justamente devido a essa limitação.

Porém, uma forma derivada desse ataque, denominada *branch prediction analysis* (análise de preditor de salto) [Aciiçmez et al. 2007], demonstrou ser possível atacar uma implementação do OpenSSL rodando em processadores convencionais (PowerPC, Intel, ARM, etc.). Executando processos maliciosos em sistemas operacionais (Windows, Linux, Android, iOS, BlackBerry, etc.), foi demonstrado ser possível afetar a execução do OpenSSL. Tornando as iterações da exponenciação modular mais lentas, passando de nanosegundos para microsegundos, foi possível detetar e inferir informações sobre a chave privada.

3.4.1.1. Unidade de predição de saltos

As instruções que compõem o código binário de um programa executável podem consumir diferentes quantidades de ciclos de *clock* de acordo com suas respectivas complexidades. Como no decorrer do fluxo de programas podem existir diversas dependências entre as instruções executadas, existe a possibilidade de que valores necessários para a execução de uma determinada instrução ainda não tenham sido calculados.

Quando a instrução depende de um salto condicional, essa situação é denominada *control hazard*. Para que o processador não permaneça ocioso até que o fluxo do programa seja definido, durante o período de decisão especula-se qual deverá ser a próxima instrução executada. Se a predição se mostrar correta (*hit*), o fluxo do programa prossegue sem degradação de desempenho; caso a predição se mostre incorreta (*miss prediction*), o *pipeline* deve ser esvaziado e a instrução correta executada. Observe que uma *miss prediction* acarreta em uma penalidade de ciclos de *clock* que é proporcional à quantidade de

estágios do *pipeline*.

Quando a CPU determina um salto como tomado (feito), ela deve buscar a instrução do endereço-alvo do salto na memória e entregá-la à unidade de execução. Para tornar o processo mais eficiente, a CPU mantém um registro dos saltos executados anteriormente no BTB (*Branch Target Buffer*). Observe que o tamanho do BTB é limitado; logo, alguns endereços armazenados precisam ser removidos para que novos endereços sejam armazenados. O preditor também possui uma parte denominada BHR (*Branch History Registers*) responsável por gravar a história dos registradores usados globalmente e localmente pelo programa [Jean-Pierre et al. 2006].

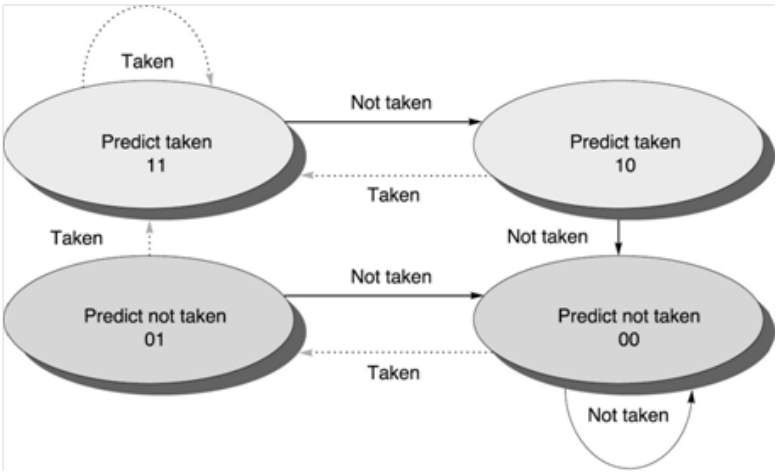


Figura 3.7: Autômato finito descreve o comportamento do preditor de saltos [Hennessy and Patterson 2002].

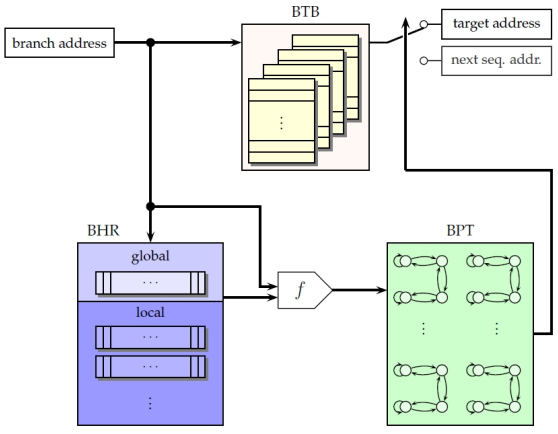


Figura 3.8: Unidade de predição de saltos [Jean-Pierre et al. 2006].

3.4.1.2. Medição direta de tempo

A máquina de estados que descreve as possíveis decisões da BTU possui um número finito de estados; logo, o algoritmo que a descreve é determinístico. O adversário pode

supor que a implementação do RSA utilizou S&M (*Square-and-Multiply exponentiation algorithm*) e MM (*Montgomery Multiplication algorithm* [Hankerson et al. 2004, Denis 2006]) e o BTU possui um autômato finito de apenas dois estados: salto tomado ou não tomado.

Seja d a chave privada, e vamos supor que o adversário conheça seus i primeiros bits e está tentando determinar d_i. Para qualquer mensagem m, o adversário pode simular as primeiras i iterações e obter um resultado intermediário que será a entrada da $(i+1)-$*ésima* iteração. Então, ele gera quatro conjuntos distintos tais que:

$$\begin{aligned}
M_1 &= \{m \mid d_i = 1 \rightarrow m \text{ causa } \textit{misspredict}\textit{ion} \text{ durante } MM\} \\
M_2 &= \{m \mid d_i = 1 \rightarrow m \text{ causa } \textit{hit} \text{ durante } MM \quad\} \\
M_3 &= \{m \mid d_i = 0 \rightarrow m \text{ causa } \textit{misspredict}\textit{ion} \text{ durante } MM\} \\
M_4 &= \{m \mid d_i = 0 \rightarrow m \text{ causa } \textit{hit} \text{ durante } MM \quad\}
\end{aligned}$$

O adversário calcula o tempo médio de execução na multiplicação de Montgomery em cada conjunto M_i. Sendo $d_i = t, t \in \{0,1\}$, então a diferença dos tempos médios de execução para o mesmo valor correto t serão muito mais significativas do que a obtida dos outros dois conjuntos, pois, para o valor incorreto, os valores de tempo de cada multiplicação terão um caráter aleatório. Portanto, se a diferença entre os tempos médios de M_1 e M_2 for muito mais significativa do que M_3 e M_4, então o palpite correto é $d_i = 1$, e $d_i = 0$ caso contrário.

Nesse ataque o adversário precisa saber de antemão o estado do BPU antes do algoritmo de decriptação ser iniciado. Uma possibilidade simples de implementação, porém menos eficiente, seria realizar a análise supondo cada um dos quatro estados iniciais. A segunda abordagem consiste em forçar o estado inicial do BPU de modo que nenhum endereço de salto esteja no BTB. Essa abordagem será fundamentalmente a mesma utilizada em todos os ataques de predição de salto listados a seguir.

3.4.1.3. Forçando BPU à mesma predição assincronamente

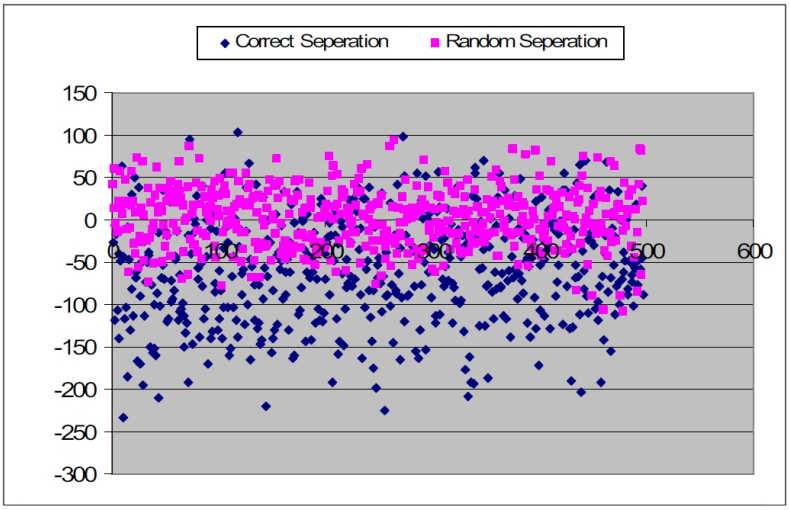
Unidades de processamento que permitem execução concorrente de processos (SMT ou *Simultaneous Multi-Threading* [Silberschatz et al. 2004]) permitem que um adversário execute um processo espião simultaneamente ao programa de encriptação. Dessa forma, o adversário pode fazer com que o valor previsto dos saltos do encriptador nunca estejam no BTB; consequentemente, sempre ocorrerá uma *misprediction* quando o resultado correto, segundo a previsão, seria que o salto fosse tomado. Comparado ao processo anterior, a análise diferencial seria similiar exceto pelo fato de que $d_i = 1$ em caso de *hit* e $d_i = 0$ em caso de *misprediction* durante o cálculo de $m^2 \bmod N$.

O processo espião remove do BTB o endereço-alvo de salto das seguintes maneiras:

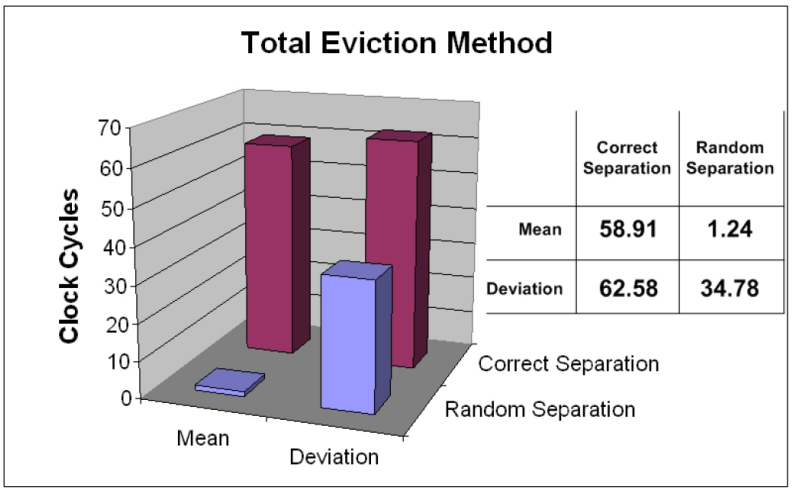
1. (*Total Eviction Method*: todas as entradas do BTB são removidas.

2. (*Partial Eviction Method*): um conjunto de entradas do BTB é removido.
3. (*Single Eviction Method*): apenas o endereço de interesse é removido da tabela.

Obviamente o primeiro método é o de implementação mais simples (assumindo que sejamos capazes de esvaziar todo o BTB entre duas iterações da exponenciação). O diferencial desse ataque é o adversário não ter que saber detalhes de implementação da BPU para ser capaz de criar o processo espião e determinar quais são os bits da chave secreta.



(a) Separações corretas e aleatórias



(b) Maior diferença das médias

Figura 3.9: Resultados práticos do *Total Eviction Method* [Jean-Pierre et al. 2006].

Esse ataque foi aplicado sobre uma implementação do RSA em OpenSSL versão 0.9.7, rodando sob uma workstation RedHat 3. Foram gerados 10 milhões de blocos de mensagens aleatórias e chaves aleatórias de 512 bits. As mensagens foram encriptadas e separadas segundo os critérios acima, sendo assumido como tomado o salto do próximo bit desconhecido.

Na Figura 3.9 (a), o eixo x corresponde aos bits do expoente de 2 até 511, sendo que cada coordenada x_i apresenta os valores das médias das separações corretas e a média das separações aleatórias, denotadas respectivamente por μ_{Y_i} e μ_{X_i}. Analisando todos os pares (μ_{Y_i}, μ_{X_i}), o adversário verifica qual deles teve a diferença mais significativa (Figura 3.9 (b)) e utiliza seus respectivos desvios padrões para determinar o desvio da diferença das médias

$$\mu_Z = \mu_Y - \mu_X = 58.91 - 1.24 = 57.67$$
$$\sigma_Z = \sqrt{\sigma_Y^2 + \sigma_X^2} = \sqrt{62.58^2 - (34.78)^2} = 71.60$$

Sempre que o adversário encontrar $Z > 0$, ele irá supor que seu palpite do valor do *bit* foi correta. O grau de certeza que o adversário pode ter nessas decisões pode ser medido pela probabilidade

$$Pr[Z > 0] = \phi(\frac{0 - \mu_Z}{\sigma_Z}) = \phi(-0.805) = 0.79.$$

Portanto, a probabilidade de suas decisões estarem corretas para essas medidas é de quase 80%, possibilitando ao adversário obter o restante da chave por força bruta.

3.4.2. Ataques de potência

Em um ataque no canal lateral de potência, o adversário analisa sutis variações no consumo de energia elétrica de um dispositivo cujo *hardware* implementa um algoritmo criptográfico (sensores RFID, *smartcards*, *SIM cards*, etc).

Operações com dados sensíveis geram alterações na corrente ou tensão da alimentação do dispositivo, permitindo extrair parcialmente (ou mesmo integralmente) a chave criptográfica e outras informações sensíveis. O primeiro ataque dessa natureza foi apresentado por [Kocher et al. 1999], também autor da célebre pesquisa precursora sobre *time attacks* [Kocher 1996].

3.4.2.1. Ataque de potência simples (SPA)

A tecnologia de semicondutores dominante em microprocessadores, memórias e dispositivos embarcados é a CMOS [Sedra and Smith 1997], sendo inversores lógicos sua unidade básica de construção. Como dispositivos utilizam fontes constantes de tensão, a potência consumida varia de acordo com o fluxo de sinais nos componentes, e esses de acordo com as operações realizadas. Se esse consumo de potência for monitorado com auxílio de um osciloscópio, poderemos estabelecer um rastro de consumo de potência (*power trace*) a cada ciclo do dispositivo.

3.4.2.2. Análise simples de potência sobre ECDSA

Uma das rotinas mais executadas em dispositivos que utilizam ECC são os algoritmos de assinatura digital de curvas elípticas (*ECDSA* ou *Elliptic Curve Digital Signature Algorithm*), tendo como operação central a multiplicação de um ponto por um escalar (Algoritmo 9).

Algoritmo 9 Binary NAF method for scalar multiplication

Entrada: $P \in E(\mathbb{F}_p), k \in \mathbb{N}$
Saída: $Q = kP \in E(\mathbb{F}_p)$
1: $(k_{t-1}, k_{t-2}, ..., k_1, k_0) \leftarrow NAF(k)$
2: $Q \leftarrow \infty$
3: **for** $j \leftarrow t-2$ **to** 0 **do**
4: $\quad Q \leftarrow 2Q$
5: $\quad$ **if** $k_i = 1$ **then**
6: $\quad\quad Q \leftarrow Q + P$
7: $\quad$ **end if**
8: $\quad$ **if** $k_i = -1$ **then**
9: $\quad\quad Q \leftarrow Q - P$
10: $\quad$ **end if**
11: **end for**
12: **return** Q

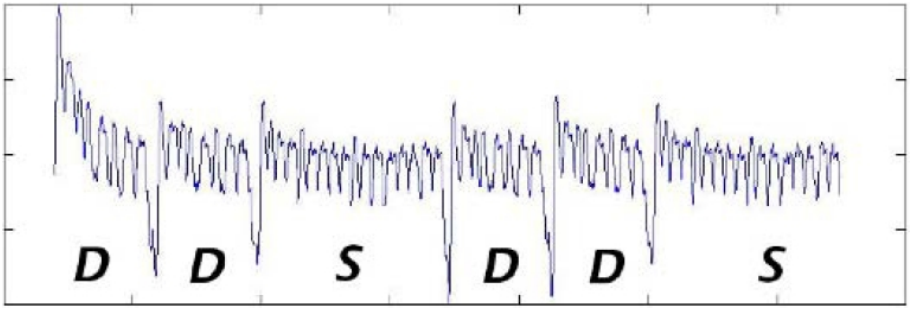


Figura 3.10: Consumo de potência durante cálculo de kP [Hankerson et al. 2004].

O que torna a forma não adjacente de k mais interessante do que sua representação binária é o fato da $NAF(k)$ possuir apenas $1/3$ de dígitos não nulos. Consequentemente uma quantidade muito menor de adições (linhas 6 e 9 do Algoritmo 9) são efetuadas.

Entretanto um adversário que soubesse que o dispositivo implementa um algoritmo *ECDSA* poderia monitorar o consumo de potência utilizando um osciloscópio, obtendo o gráfico mostrado na Figura 3.10. No Algoritmo 9, vemos que adições são realizadas apenas quando $k_i \neq 0$; logo, uma maior quantidade de potência é despendida para dígitos não nulos. Portanto os intervalos curtos denominados D correspondem a iterações em que $k_i = 0$, enquanto intervalos longos denominados S correspondem a iterações em que $k_i \neq 0$. Essa informação torna viável descobrir a chave por meio de ataques de força bruta, pois apenas $1/3$ dos dígitos são não nulos.

A solução mais simples contra SPA consiste em inserir operações redundantes no algoritmo de multiplicação (Algoritmo 10), de modo que a sequência de operações elementares envolvidas sejam realizadas em igual proporção. Comparando o novo *power trace* obtido (Figura 3.11) não é possível diferenciar adições de multiplicações.

Algoritmo 10 Binary NAF method for scalar multiplication resistant to SPA

Entrada: $P \in E(\mathbb{F}_p), k \in \mathbb{N}$
Saída: $Q = kP \in E(\mathbb{F}_p)$
1: $(k_{t-1}, k_{t-2}, ..., k_1, k_0) \leftarrow NAF(k)$
2: $Q \leftarrow \infty$
3: **for** $i = t-1$ **to** 0 **do**
4: $\quad Q_0 = 2Q_0$
5: $\quad Q_1 = Q_0 + P$
6: $\quad Q_0 = Q_{k_i}$
7: **end for**
8: **return** Q

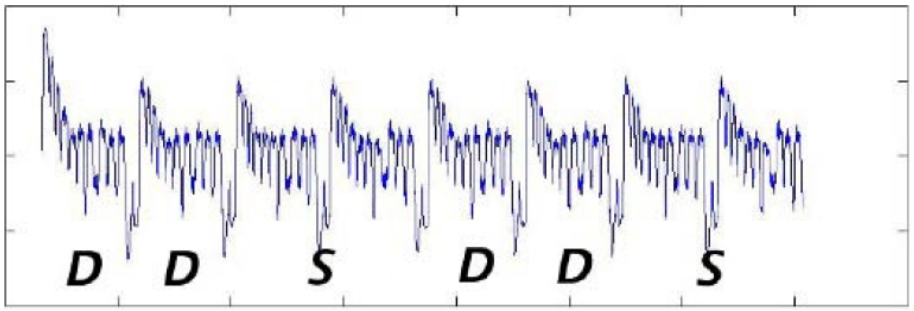


Figura 3.11: Consumo de potência durante cálculo de kP [Hankerson et al. 2004].

3.4.2.3. Ataque de potência diferencial (DPA)

Quando a variação do consumo de potência não é sensível o suficiente em relação às operações executadas por um dispositivo, o adversário pode monitorar como o consumo se comporta em relação ao valor de uma determinada variável. Nesse ataque, primeiramente detetamos uma variável V, influenciada durante um processo de decriptação ou assinatura digital por um texto m e uma porção desconhecida da chave privada. A partir disso, definimos a função de seleção $V = f(k', m)$.

O adversário então coleta milhares de *power traces*, determinando indutivamente todos os bits que compõem a chave privada através do cálculo da derivada dessa função. Para cada bit k'_i corretamente previsto obtemos uma derivada não nula para os valores de k' e m, caso contrário a derivada é nula. O processo é repetido até que cada k'_i seja determinando [Hankerson et al. 2004]. Esse modelo de ataque é conhecido como Análise Diferencial de Potência (DPA ou *Differential Power Analysis*).

3.4.2.4. Análise diferencial de potência sobre ECDSA

Ainda que o Algoritmo 10 tenha sido adotado, podemos aplicar um DPA sobre o processo de ECDSA.

Determinada uma variável V cujo valor influencie o consumo de potência, e uma função de seleção f tal que $V = f(k', m)$, o adversário coleta milhares de *power traces*, estima o tamanho que a porção k' ocupa na chave privada e separa os dados coletados em dois grupos de acordo com o valor previsto de V.

Suponha que o adversário colete os *power traces* durante os cálculos de $kP_1, kP_2, ..., kP_r$, para os quais foi usado o Algoritmo 4 (*Double-and-add always*), que repetimos aqui para facilitar a leitura.

Algoritmo 4 Double-and-add-always

Entrada: $P \in E(\mathbb{F}_p), k = (k_{t-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Saída: $kP \in E(\mathbb{F}_p)$
1: $R_0 \leftarrow P$
2: **for** i from $t-2$ **downto** 0 **do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad R_1 \leftarrow R_0 + P$
5: $\quad b \leftarrow k_i$
6: $\quad R_0 \leftarrow R_b$
7: **end for**
8: **return** R_0

Como $P_1, P_2, ..., P_r$ são públicos, ele precisa determinar apenas k. Dado $R_0 = \infty$,

	R_0	R_0	k_{t-1}	$R_0 \leftarrow R_{k_{t-1}}$
1	∞	P	1	P
2	...	...	...	...
3	...	...	...	...
...	...	...	...	...

Tabela 3.1: $k = (1, k_{t-2}, k_{t-3}, ..., k_1, k_0)$

o passo 3 do Algoritmo 4 é trivial e pode ser distinguido de uma operação não trivial pelo *power trace*; logo, o adversário pode facilmente identificar o bit mais a esquerda cujo valor é 1. Tomando $k_{t-1} = 1$, na segunda iteração do algoritmo temos que $R_0 = 2P$ (se $k_{t-2} = 0$) ou $R_0 = 3P$ (se $k_{t-2} = 1$). Consequentemente, na terceira iteração, o valor $4P$ ser computado apenas se $k_{t-2} = 0$. Definindo $k' = k_{t-2}$ e $m = P_i$ (i-ésimo bit do ponto $4P = (4P_1, 4P_2, ..., 4P_i, ..., 4P_r,)$), a função seletora calcula o valor do bit $4P_i$. Se o gráfico do consumo de potência da função apresentar picos, então $k_{t-2} = 0$, caso contrário $k_{t-2} = 1$. Esse processo é repetido até todos os bits de k serem determinados [Hankerson et al. 2004].

	R_0	R_0	k_{t-1}	$R_0 \leftarrow R_{k_{t-1}}$
1	∞	P	1	P
2	$2P$	$4P$	?	?
3	...	...	...	...
...	...	...	...	...

Tabela 3.2: $k = (1, k_{t-2}, k_{t-3}, ..., k_1, k_0)$

	R_0	R_0	k_{t-1}	$R_0 \leftarrow R_{k_{t-1}}$
1	∞	P	1	P
2	$2P$	$4P$	0	$2P$
3	$4P$	$6P$	...	...
...	...	...	...	...

Tabela 3.3: $k = (1, 0, k_{t-3}, ..., k_1, k_0)$

Se a curva elíptica for gerada sobre um $\mathbb{F}_p$ de característica superior a 3, podemos usar um sistema misto de representação de coordenadas no qual P seja representado em um sistema de coordenadas afins, enquanto R_0 e R_1 são representados em coordenadas jacobianas [Hankerson et al. 2004].

Se $P = (x, y)$ no sistema afim, após a primeira atribuição $R_1 \leftarrow P$ teríamos $R_1 = (x : y : 1)$. Então, R_1 seria aleatorizado com $(\lambda^2 x, \lambda^3 y, \lambda)$ e o algoritmo procederia como o usual. Desse modo o adversário estaria impedido de realizar predições baseadas no valor de um bit específico $4P_i$ em sistemas de coordenadas jacobianas aleatorizadas.

3.4.2.5. *High-order DPA*

Uma variação do ataque de potência diferencial é o *high-order* DPA (HODPA), que é baseado em um conjunto de propriedades estatísticas do sinal de potência. Para obter uma melhor taxa de acerto dos *bits* da chave, é necessária uma quantidade maior de amostras. Além disso, existe uma complexidade maior durante a implementação desse ataque [Gierlichs et al. 2009].

Um ponto importante a ser observado é que esse ataque possui a capacidade de se sobrepor à resistência de uma contramedida de *masking* e obter os *bits* da chave. Isso é possível pela análise tanto do valor mascarado como da própria máscara criando uma relação entre ambos. Um problema encontrado na utilização desse ataque é a identificação do momento em que é necessário obter o sinal referente ao valor mascarado ou a máscara.

3.4.2.6. *Template attacks*

O *Template Attack* (TA) utiliza amostras de potência elétrica para executar uma análise, na tentativa de obter os *bits* da chave privada. É considerado o mais poderoso dentre os

ataques de potência diferencial. Existe uma divisão em duas fases do ataque, a saber, a fase de construção e a fase de correlação. A primeira gera um *template* de um padrão obtido no dispositivo alvo. Na segunda é analisada a correlação entre o *template* gerado e as amostras obtidas durante a execução do dispositivo [Özgen et al. 2016]. Esse ataque pode ser combinado com algoritmos de classificação para diferenciar valores da chave daqueles do restante, como é proposto no trabalho de [Özgen et al. 2016]. Uma outra opção é a utilização de algoritmos de reconhecimento de padrão com aprendizado de máquina.

3.5. Ataques e contramedidas para criptografia simétrica

Implementações inseguras de criptografia simétrica podem ser suscetíveis a diversos tipos de ataques de canal lateral. Mesmo implementações de algoritmos de alto desempenho podem ser vulneráveis a canais laterais de tempo executados localmente (via latência da memória *cache*) ou remotamente (tempo de resposta na comunicação em rede). Contramedidas comuns para criptografia simétrica contra ataques de canal lateral envolvem execução em tempo constante e mascaramento.

3.5.1. Ataques por canal lateral de tempo e contramedidas

Um ataque simples contra a utilização de criptografia simétrica envolve a aplicação de funções de resumo criptográfico para autenticação. Nessas aplicações, é comum que o algoritmo simétrico produza um autenticador ou *hash* das credenciais que precisa ser comparado com o resultado correto. Durante a comparação de cadeias de caracteres, pequenas variações no tempo de execução podem fornecer informação útil para o adversário reduzir enormemente a complexidade de um ataque de busca exaustiva. Ao comparar cadeias, se a interrupção do laço acontecer exatamente na primeira posição diferente, o adversário é capaz de realizar um ataque de busca exaustiva em cada *byte* individualmente, baseado no tempo de resposta. Desta forma, é possível determinar o autenticador de uma mensagem ou *hash* de uma senha, por exemplo, sem conhecimento de segredos. O trecho de código abaixo ilustra uma forma segura de comparação, cujo tempo de execução é invariante e saturado para o pior caso:

```
int cmp_const(const void * a, const void * b,
              const size_t size) {
    const unsigned char *_a = a, *_b = b;
    unsigned char result = 0;
    size_t i;
    for (i = 0; i < size; i++) {
        result |= _a[i] ^ _b[i];
    }
    return result;
}
```

Desvios condicionais dentro de algoritmos criptográficos podem ser outra fonte de problemas, especialmente em processadores modernos com execução especulativa e predição de desvios. A remoção de desvios condicionais envolve a aplicação de técnicas

de programação sem desvios condicionais. A aplicação correta dessas técnicas é altamente dependente do algoritmo sendo estudado, mas uma generalização útil é calcular os dois ramos do desvio condicional simultaneamente e utilizar operações mascaradas para selecionar o valor correto apenas ao final da execução. A ideia é ilustrada pelo trecho de código abaixo para seleção entre dois valores em tempo constante, dependendo de um bit de condição:

```
unsigned select(unsigned a, unsigned b,
                unsigned bit) {
  /* -0 = 0, -1 = 0xFF....FF */
  unsigned mask = - bit;
  unsigned ret = mask & (a^b);
  ret = ret ^ a;
  return ret;
}
```

Outra possibilidade é utilizar uma variante da função de seleção para alterar em tempo constante os valores de entrada do trecho de código protegido ser executado.

Acessos à memória devem também ser cuidadosamente protegidos. No contexto de cifras de bloco, a preocupação se concentra na representação de caixas de substituição como vetores ou introdução de qualquer tabela pré-calculada para acelerar operações sobre bits realizadas pela cifra. O algoritmo AES ilustra muito bem o problema, pois seu desempenho depende enormemente da eficiência das caixas de substituição, motivando o implementador a utilizar tabelas simples. Entretanto, um adversário capaz de monitorar o comportamento da memória *cache* pode determinar que porções da caixa de substituição são usadas na etapa de cifração e recuperar informação sensível [Bernstein 2004, Percival 2005, Bonneau and Mironov 2006, Tromer et al. 2010]. Por exemplo, um ataque *Flush and Reload* utiliza a instrução `clflush` em processadores Intel para eliminar endereços de páginas compartilhados da memória *cache* e verifica se os dados retornam à cache após permitir que o programa atacado execute um pequeno número de instruções [Yarom and Falkner 2014]. Por funcionar no nível mais baixo e compartilhado da memória *cache*, o ataque torna-se possível entre núcleos distintos e contra ambientes virtualizados.

Existem contramedidas de diversos tipos para mitigar o problema. Uma opção simples é adotar arquiteturas com latência de acesso uniforme à memória, como alguns microcontroladores simples. Outra possibilidade é utilizar uma implementação em hardware do algoritmo, quando disponível, que deve oferecer uma superfície de ataque menor. Alternativas mais sofisticadas para implementação em software são *bitslicing*, onde as operações sobre bits são realizadas explicitamente, sem ajuda de tabelas pré-calculadas, e o impacto em desempenho é reduzido pela aplicação do mesmo circuito lógico a bits de diferentes variáveis simultaneamente [Biham 1997, Käsper and Schwabe 2009]. Para tabelas pequenas, também é possível utilizar uma instrução para acesso a tabela armazenada em um registrador [Hamburg 2009] ou percorrer a tabela inteira a cada leitura, utilizando a função de seleção apresentada anteriormente para realizar uma cópia condicional entre o valor lido e um valor atual da variável de interesse.

3.5.2. Ataques de potência e contramedidas

Ataques de potência também são possíveis contra implementações de primitivas simétricas, por exemplo algoritmos como o HMAC para autenticação de mensagens (MAC) utilizando uma função de resumo criptográfico H como fonte de segurança. Para uma chave simétrica k, o HMAC calcula o autenticador para M como

$$\text{HMAC}_k(M) = H((k \oplus opad) \parallel H((k \oplus ipad) \parallel M)),$$

onde $ipad$ é a cadeia produzida pela repetição do valor `0x36` e $opad$ pela repetição do valor `0x5C`. Ataques diferenciais de potência foram demonstrados contra implementações inseguras do HMAC utilizando tanto funções de resumo construídas segundo o paradigma Merkle-Damgård quanto aquelas baseadas em cifras de bloco.

A característica principal de um ataque diferencial de potência é que, em um certo ponto na execução do algoritmo, uma variável combina conhecimento de uma função f sobre um valor secreto fixo com outro valor conhecido pelo adversário. O adversário pode então formular hipóteses sobre o valor secreto fixo e aplicar a função de acordo com um certo modelo de vazamento. Pela captura de traços de consumo de energia, é possível verificar a validade das hipóteses. Supondo que o adversário possua controle do dispositivo, obtenha informação sobre a distância de Hamming entre estados internos consecutivos durante o cálculo instâncias de HMAC, e conheça as mensagens cuja autenticação é verificada, o objetivo é combinar informação fixa desconhecida com informação variável conhecida.

O ataque proposto por [McEvoy et al. 2007] se concentra na primeira execução da função de resumo interna $H(k \oplus ipad)$ ao HMAC, que é invariante em execuções distintas. Esse cálculo irá produzir um valor de resumo intermediário, e o processamento prossegue com a mensagem conhecida M. Portanto, o objetivo do ataque é recuperar o valor intermediário pela formulação de hipóteses verificadas após o processamento de M. Na fase final, o ataque é repetido na função de resumo externa do HMAC, permitindo a forja de autenticadores para mensagens da escolha do atacante, sem conhecimento da chave criptográfica. Um ataque similar pode ser montado contra funções de resumo construídas a partir de cifras de bloco, mesmo quando a cifra de bloco é ideal e resistente contra ataques de canal lateral. [Okeya 2006] demonstrou ser possível recuperar a chave criptográfica completa nesse caso, permitindo forja de autenticadores para mensagens escolhidas pelo adversário.

Ataques cúbicos [Dinur and Shamir 2012] são ataques genéricos de recuperação de chaves que podem ser aplicados automaticamente a qualquer criptossistema em que um único bit de informação pode ser representado por um polinômio de grau pequeno na chave e texto claro, como cifras de fluxo baseadas em registradores de deslocamento. O ataque consiste em somar o bit de saída de todos os possíveis valores de um subconjunto de bits públicos de entrada, escolhidos de forma que o bit resultante seja uma combinação linear de bits secretos. Aplicações repetidas da mesma técnica produzem relações lineares entre bits secretos que podem ser resolvidas para descobrir esses bits. Os autores demonstram que bits públicos dessa forma existem com alta probabilidade quando a cifra aproxima um polinômio aleatório de grau suficientemente pequeno e podem ser encontrados em uma fase de pré-computação. A versão de canal lateral do ataque funciona por

meio da captura de bits individuais que satisfazem as características do ataque, tipicamente por um ataque de potência.

Mascaramento

Mascaramento é uma das contramedidas contra ataques de potência mais investigadas na literatura para proteger valores sensíveis, como texto claro durante a encriptação ou criptogramas durante a decriptação. Como a informação calculada durante esses processos se tornará a saída dos algoritmos, todos os valores intermediários precisam ser protegidos durante todo o tempo. Ao se aplicar mascaramento, um conjunto de n máscaras $m_0, \ldots, m_n$ pseudo-aleatórias é utilizado por meio da operação `XOR` para representar valores intermediários, de forma que a informação vazada por canais laterais não mais está correlacionada com informação secreta legítima. O valor mascarado m com $d+1$ máscaras é dado por $m = \bigoplus_{i=0}^{d} m_i = m_0 \oplus m_1 \oplus \ldots \oplus m_d$. As operações em um corpo binário podem então ser adaptadas para funcionar com valores mascarados:

1. Toda operação linear L sobre um valor mascarado consiste em aplicar a mesma operação sobre as máscaras:

$$L(m) \equiv L(m_0 \oplus m_1 \oplus \ldots \oplus m_d) \equiv L(m_0) \oplus L(m_1) \oplus \ldots \oplus L(m_d)$$

2. Uma operação `NOT` pode ser calculada como:

$$\neg m \equiv \neg m_0 \oplus m_1 \oplus \ldots \oplus m_d$$

3. Uma operação `XOR` entre valores mascarados $a = \bigoplus_{i=0}^{d} a_i$ e $b = \bigoplus_{i=0}^{d} b_i$ pode ser calculada como:

$$a \oplus b \equiv \bigoplus_{i=0}^{d} a_i \oplus \bigoplus_{i=0}^{d} b_i \equiv \bigoplus_{i=0}^{d} (a_i \oplus b_i)$$

4. Uma operação `AND` entre $a = \bigoplus_{i=0}^{d} a_i$ e $b = \bigoplus_{i=0}^{d} b_i$ é mais complicada e exige um algoritmo específico (Algoritmo 11) [Ishai et al. 2003].

Algoritmo 11 Operação `AND` aplicada sobre valores mascarados a e b.

Entrada: Valores (a_i) e (b_i) tais que $\oplus_{i=0}^{d} a_i = a$ e $\oplus_{i=0}^{d} b_i = b$.
Saída: Valores (c_i) tais que $\oplus_{i=0}^{d} c_i = a \wedge b$

```
1:  for i from 0 to d do
2:      r_{i,i} ← 0;
3:      for j from i + 1 to d do
4:          r_{i,j} ← random();
5:          r_{j,i} ← (r_{i,j} ⊕ (a_i ∧ b_j)) ⊕ (a_j ∧ b_i);
6:      end for
7:  end for
8:  for i from 0 to d do
9:      c_i ← a_i ∧ b_i;
10:     for j from 0 to d do
11:         c_i ← c_i ⊕ r_{i,j};
12:     end for
13: end for
```

Estas observações permitem que qualquer algoritmo utilizando operações binárias seja implementado de maneira mascarada. A contramedida causa baixa penalidade em desempenho quando aplicada sobre sequências de operações binárias lineares, mas o impacto aumenta consideravelmente sobre operações binárias não-lineares, que exigem formulação distinta da computação.

3.5.3. Oráculo de preenchimento

Em um ataque de oráculo de preenchimento, um servidor (oráculo) vaza informação sobre quanto do preenchimento de uma mensagem está no formato correto. O adversário entào manipula partes do criptograma para conseguir decriptar (ou algumas vezes encriptar) mensagens utilizando a chave criptográfico do oráculo, mesmo sem conhecê-la. Por se tratar de um ataque ativo, a forma mais simples de mitigá-lo é autenticar criptogramas utilizando um algoritmo de MAC, com verificação realizada em tempo constante. O ataque original foi proposto por Vaudenay [Vaudenay 2002], mas implementações de protocolos criptográficos em que o ataque foi mitigado mostraram-se posteriormente vulneráveis à versão do ataque no canal lateral de tempo *Lucky Thirteen* [AlFardan and Paterson 2013].

Em criptografia simétrica, o ataque é particularamente efetivo contra o modo de operação CBC para cifras de bloco. Suponha que o atacante tenha capturado o vetor de inicialização IV e três blocos de criptograma C_1, C_2, C_3, e queira decriptar o segundo bloco para recuperar M_2, sabendo que C_3 possui preenchimento correto. Se o adversário manipula o último byte de C_1 e submete (IV, C_1, C_2) para o servidor, a decriptação deverá alterar completamente o resultado de M_1 e o último byte de M_2. O servidor então verifica o preenchimento correto de M_2 e retorna essa informação para o adversário. Para descobrir o último byte de M_1, basta somar ao último byte de C_1 uma estimativa do último byte de M_2 e o byte correto de preenchimento, de forma que a ausência de erro de preenchimento indica que a estimativa estava correta. Após descobrir o último byte de M_2 com várias tentativas, o ataque pode continuar sobre os bytes individuais restantes do bloco.

3.6. Ataques e contramedidas para ECC

3.6.1. Níveis em que os ataques SCA podem ser aplicados

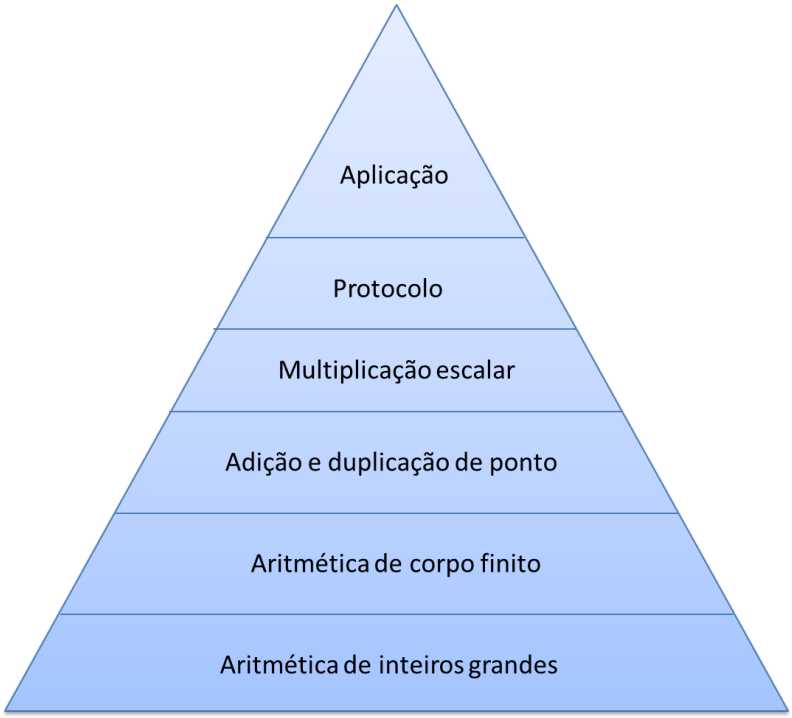


Figura 3.12: Pirâmide de implementação de criptografia baseada em curvas elípticas (ECC). Qualquer uma destas camadas pode ser vulnerável a ataques por canais laterais.

A implementação de protocolos criptográficos baseados em curvas elípticas depende da existência de implementações de um conjunto de operações básicas: multiplicação de ponto por escalar, adição e duplicação de ponto, aritmética de corpo finito e aritmética de inteiros grandes. Protocolos criptográficos são implementados a partir dessas primitivas e, então, utilizados em aplicações as mais variadas. A Figura 3.12 ilustra tais operações em camadas, onde a implementação de uma camada depende apenas da existência de funcionalidades na camada imediatamente inferior.

A representação em camadas, entretanto, esconde a interdependência que existe entre operações. Na prática, em bibliotecas de primitivas e protocolos criptográficos, há interações entre níveis não adjacentes, pois tais interações são relevantes para desempenho e, em alguns casos, segurança.

Em uma implementação de uma dada aplicação, qualquer um dos níveis da pirâmide pode estar vulnerável a ataques por canais laterais. Com relação ao canal de tempo, por exemplo, se uma operação não é de tempo constante com relação à chave ou valores intermediários dependentes da chave, então todas as operações em níveis superiores vazam informações por diferenças potencialmente observáveis de tempo.

3.6.2. Ataques de tempo e SPA ao algoritmo double-and-add-not-always

3.6.2.1. Ataque de tempo ao algoritmo double-and-add-not-always

Considerando que o SPA de tempo se baseia na variação de tempo do algoritmo relativamente ao valor da chave, o sinal mais simples de que uma implementação é insegura são desvios condicionais (*if*s) cujo comportamento dependa apenas de *bits* da chave privada.

Nesse contexto, o ataque de tempo aplicado a uma implementação do RSA de Kocher [Kocher 1996] pode ser aplicado a implementações de curvas elípticas: tudo o que faz o adversário é medir o tempo de execução do algoritmo, advinhar os *bits* da chave, e validar o resultado testando a chave numa operação de decriptação.

No trabalho de [Danger et al. 2013] é descrito um ataque a uma implementação genérica de ECSM, que segue os seguintes passos: primeiramente o adversário coleta o tempo de execução de diferentes ECSMs com o mesmo escalar e diferentes pontos base. Para cada ECSM, simula a computação usando um simulador de software com exatamente a mesma implementação do chip alvo, "chutando"o valor do bit i do escalar.

Agora suponha, sem perda de generalidade, que a hipótese é de que o valor do bit seja zero. O adversário então separa os diferentes tempos de execução em dois conjuntos, S_1 e S_2. Caso uma redução seja necessária ao final da execução, o tempo obtido é armazenado em S_2, senão é armazenado em S_1. Após toda a execução, é feita uma média dos tempos armazenados em S_1 e S_2 e, se a diferença entre as médias for aproximadamente o tempo de execução da redução, a hipótese estava correta.

3.6.2.2. Ataque SPA ao algoritmo double-and-add-not-always

O algoritmo *double-and-add-not-always* executa em tempo constante; contudo, o ataque SPA (com ou sem power model) pode ser aplicado para distinguir os padrões no *trace* das iterações com apenas DBL (onde o *bit* da chave é igual a 0) daquelas com DBL+ADD (com *bit* da chave igual a1). Um ataque deste tipo segmenta/divide o *trace* de potência de uma execução do ECSM em *subtraces*, cada uma contendo uma operação de ponto (ADD ou DBL).

Se o tempo de execução de uma operação de ADD for diferente do tempo de DBL, então o comprimento dos *subtraces* revela onde estão os ADDs e consequentemente os bits do escalar. Se o tempo da operação ADD e o tempo da DBL forem iguais, e uma fórmula unificada para ADD and DBL é utilizada, então, se for aplicada correlação (coeficiente de correlação de Pearson) entre todos os pares de *subtraces*, o resultado será que a correlação será mais alta para os pares de *subtraces* cuja operação correspondente é a mesma (i.e., (ADD,ADD) ou (DBL,DBL)), identificando portanto as operações de ponto e consequentemente os *bits* do escalar.

Mesmo que a implementação seja vulnerável a SPA é desejável que execute em tempo constante, o que cria uma base para a implementação de outras contramedidas.

3.6.3. Algoritmo Double-and-add-always de Coron [Coron 1999]

O algoritmo *double-and-add-always* de [Coron 1999] (Algoritmo 4) utiliza uma adição falsa de ponto conhecida como *dummy point addition*, quando o bit escalar k_i é 0, tornando a sequência de operações computadas durante a multiplicação escalar independente do valor do escalar.

Portanto, o adversário não consegue, em princípio, adivinhar o valor do *bit* k_i por SPA. Uma desvantagem desse método é a sua baixa eficiência. Ele requer $nA + nD$ operações no corpo, um aumento de 33% nas operações do corpo em comparação com a versão desprotegida binária do algoritmo *left-to-right*.

3.6.3.1. Ataque de duplicação de Fouque e Valette [Fouque and Valette 2003]

O ataque de duplicação de Fouque-Valette [Fouque and Valette 2003] é baseado no fato de que é possível detetar se dois valores intermediários são iguais quando o algoritmo calcula a multiplicação escalar para pontos escolhidos P e $2P$. Diversos algoritmos protegidos contra SPA são vulneráveis ao ataque de Fouque e Valette, como o algoritmo clássico *left-to-right* binário, incluindo as variações do mesmo, como o *double-and-add-always* de Coron.

No algoritmo de Coron (Algoritmo 4), a soma parcial é calculada da seguinte maneira: $S_m(P) = \sum_{i=1}^{m} k_{n-i}2^{m-i}P = \sum_{i=1}^{m-1} k_{n-i}2^{m-1-i}(2P) + k_{n-m}P = S_{m-1}(2P) + k_{n-m}P$. Assim, o resultado intermediário do algoritmo no passo m quando a entrada for P, será igual ao resultado intermediário no passo $m-1$ quando a entrada for $2P$, se e somente se $k_{n-m} = 0$. Portanto, um adversário pode obter o escalar secreto comparando a computação de duplicação no passo $m+1$ para P e no passo m para $2P$ para recuperar o bit k_{n-m}. Se ambas as computações forem idênticas, $k_{n-m} = 0$, senão $k_{n-m} = 1$. Isso mostrou que com apenas duas requisições de multiplicação escalar escolhidas pelo adversário, é possível recuperar todos os *bits* do escalar. [1]

3.6.4. Contramedidas

3.6.4.1. Aleatorização do escalar (Scalar Randomization-SR)

Aleatorização do escalar é aplicada no início da multiplicação escalar, da seguinte maneira:

- Aleatoriamente selecione $r \in_R \{0,1\}^n$, para um valor pequeno de n. O valor $n = 32$ é uma escolha razoável, que balanceia segurança e eficiência.
- Calcule $k' \leftarrow k + r|E|$;
- Utilize k' no lugar de k.

O custo da aplicação dessa contramedida depende diretamente de: geração dos bits pseudo-aleatórios de r; n iterações da ECMS; as adições e multiplicações para calcu-

[1] O adversário coleta um traço de energia para a computação de kP e um para a computação de $k(2P)$. Para cada iteração $m = 1,\ldots,n$ ele executa o ataque como descrito e encontra k_{n-m}.

lar k' e n bits da SRAM. Já no quesito eficiência é necessário verificar que se é vulnerável a ataques de *template* online (OTA) [Batina et al. 2014], entre outros.

3.6.4.2. Realeatorização de coordenadas projetivas (Projective Coordinates (Re)randomization (CR) [Coron 1999]

As coordenadas projetivas usadas no Algoritmo Montgomery Ladder são $(X : Z)$. Os passos para aplicar essa contramedida são:

- Gere um valor pseudo-aleatório $\lambda \mathbb{F}_p \backslash \{0\}$;
- Calcule $Z_2 \leftarrow \lambda$ e $X_2 \leftarrow u \cdot \lambda$, onde u é a coordenada x do ponto P da entrada;
- Utilize $P' = (X_2 : Z_2)$ no lugar de P.

Existe um custo de geração de $\lceil log_2(p) \rceil$ bits aleatórios. Além disso, algumas vantagens ligadas a essa randomização das coordenadas, como por exemplo a resistência a ataques online de *template* [Batina et al. 2014], pode ser aplicada para as curvas de Edwards, *twisted* Edwards e Montgomery. Por fim, a cada iteração do método de ECSM ela pode ser aplicada, sendo apenas necessária a coordenada x no caso de Montgomery Ladder.

3.6.4.3. Point Blinding (PB) [Coron 1999]

Essa contramedida pode ser aplicada antes da primeira multiplicação escalar, seguindo os passos seguir:

- Gerar um ponto aleatório R no subgrupo;
- Pré-calcule e armazene $S = kR$;
- No início de cada operação escalar calcule $T \leftarrow P + R$ e $Q \leftarrow kT$;
- Atualize R e S da seguinte maneira, com t aleatório:
  - $R \leftarrow (-1)^t 2R$;
  - $S \leftarrow (-1)^t 2S$;
- Retorne $W = Q - S$.

O custo dessa contramedida é de 2 adições (ECADD) e 2 duplicações (ECDBL) elípticas, e memória SRAM para valores temporários. Provavelmente também é utilizada memória não volátil para armazenar R e S. *Point Blinding* protege contra SVA horizontal [Murdica et al. 2012], RPA [Goubin 2003], e ZPA [Akishita and Takagi 2003], mas é vulnerável a OTA [Batina et al. 2014].

3.6.4.4. Particionamento de escalar (Scalar Splitting (SS) [Clavier and Joye 2001])

A aplicação dessa contramedida segue os seguintes passos, onde k é o escalar original de n bits:

- Gere um inteiro $k_1 < k$ aleatoriamente e calcule:
  1. $k_2 \leftarrow k - k_1$;
  2. $Q \leftarrow k_1 P$;
  3. $T \leftarrow [k_2]P$;
  4. $R \leftarrow Q + T$.

O custo dessa contramedida é a execução de 1 (multiplicação escalar) ECSM de n bits e 1 adição (ECADD), que pode ser reduzido se for empregado o truque de Shamir para multiplicação escalar dupla (versão regular) [Ciet and Joye 2003]. A eficácia está ligada à resistência aos ataques TA, SPA clássico e CPA clássico. As variantes dessa contramedida são: *Euclidean splitting* [Ciet and Joye 2003] e *multiplicative splitting* [Trichina and Bellezza 2003], ambas com a mesma eficácia da versão original, também conhecida como *additive splitting*.

3.6.5. Ataques Horizontais (HA)

Ataques horizontais (HA) são uma metodologia para ataques por canal lateral, cujos alvos são as principais operações criptográficas em protocolos baseados em RSA e ECC, a exponenciação modular e a multiplicação escalar, respectivamente. Em teoria, tais ataques permitem recuperar os bits do expoente/escalar secreto através da análise de *traces* individuais, isto é, apenas um único *trace* obtido do alvo é suficiente; portanto, são eficazes contra implementações protegidas por contramedidas como SR, CR, PB e SS.

Um requisito básico dos ataques horizontais é o conhecimento do algoritmo de multiplicação escalar. De posse de tal informação, o adversário pode escolher, dentre outros, os seguintes métodos ou
emphdistinguishers: correlation analysis, collision-correlation analysis e *cluster analysis*.

O método de correlation analysis [Clavier et al. 2010] segue o mesmo princípio da análise de potência por correlação (CPA)[2] aplicada a um conjunto de *traces*, na configuração vertical. A diferença para o contexto horizontal é de que um único *trace* é dividido em vários segmentos e para cada um destes segmentos um valor intermediário hipotético é atribuído, em relação a um chute sobre o valor da chave. A correlação entre amostra e valor hipotético é calculada do mesmo modo que CPA, e os modelos de vazamentos usualmente utilizados são o peso de Hamming e a distância de Hamming. Este método funciona contra implementações protegidas somente com SR, ou quando CR é também aplicada mas o parâmetro aleatório utilizado é curto, e ataques por força bruta ao valor de tal parâmetro são viáveis.

O método de collision-correlation analysis [Bauer et al. 2013, Bauer and Jaulmes 2013, Clavier et al. 2012, Witteman et al. 2011b, Walter 2001]

[2] *Correlation power analysis*.

computa a correlação ou distância euclideana entre diferentes segmentos de um *trace*. O objetivo principal é identificar a ocorrência de um mesmo dado intermediário em diferentes partes de um *trace*, e com isso derivar os bits do escalar secreto. Para tanto, o adversário deve ter conhecimento do algoritmo de ECSM empregado. Em teoria, este método é viável contra implementações envolvendo combinações de contramedidas clássicas.

A maioria das formas de ataques horizontais requer pré-processamento avançado dos *traces*, caracterização e avaliação de vazamento antes da aplicação de distinguishers. Os principais problemas da abordagem horizontal são de que extrair o vazamento a partir de um único *trace* tipicamente apresenta fortes limitações e desafios, como o alto nível de ruído e a indisponibilidade de amostras rotuladas. Em particular, métodos de avaliação de vazamento, como o TVLA [Goodwill et al. 2011], requerem amostras rotuladas e isso não é possível quando a contramedida SR é aplicada.

Métodos baseados em aprendizado não supervisionado, mais especificamente aqueles baseados em *clustering*, foram recentemente aplicados para resolver tais limitações e têm se mostrado capazes de produzir resultados práticos. Heyszl et al [Heyszl et al. 2014] propõem aplicar classificação por *clustering* a um único *trace* para possibilitar a identificação de classes específicas de operações; este método funciona bem para medições com baixo ruído e requer uma estação de EM composta de múltiplas sondas. Perin et al. [Perin et al. 2014], consideram uma abordagem heurística baseada na diferença de médias para a seleção de pontos de interesse. Além disso, ambas soluções usam um único *trace* como entrada para a etapa de avaliação de vazamento, o qual pode ter sido muito afetado por uma grande quantidade de ruído. Perin e Chmielewski [Perin and Chmielewski 2015] fornecem uma metodologia para ataques horizontais baseados em *clustering* que foca em corrigir as deficiências dos trabalhos mencionados anteriormente.

3.6.6. Ataque HCA ao Montgomery Ladder c/ SR + CRR

3.6.6.1. Preparação: filtragem, segmentação e alinhamento

Em ataques horizontais, devido a problemas como o alto nível de ruído presente em um *trace*, fenômenos como clock drift[3] e variações no tempo em que o dispositivo de medição (osciloscópio) inicia a medição após o recebimento do sinal de trigger (*trigger jitter*), é fundamental o pré-processamento dos *traces* medidos antes de se iniciar a análise, particularmente as operações de filtragem, segmentação e alinhamento.

Filtragem. A Section 3.6.6.1 mostra um *trace* não filtrado e o mesmo *trace* após aplicação de filtro baixa, onde pode-se identificar com mais clareza características do sinal, como periodicidade e amplitude, bem como as rodadas ou iterações. É recomendável que filtros analógicos sejam aplicados, sempre que possível, de modo que o sinal que é digitalizado pelo osciloscópio contenha apenas as frequências desejadas, permitindo o uso da menor resolução (range) vertical suportada pelo dispositivo, o que pode não ocorrer caso picos

[3] *Clock drift* ou *clock jitter* é o desvio do sinal de clock real em relação ao sinal verdadeiramente periódico de referência. Devido ao clock drift, o sinal de clock real não é periódico, mas sim aproximadamente periódico.

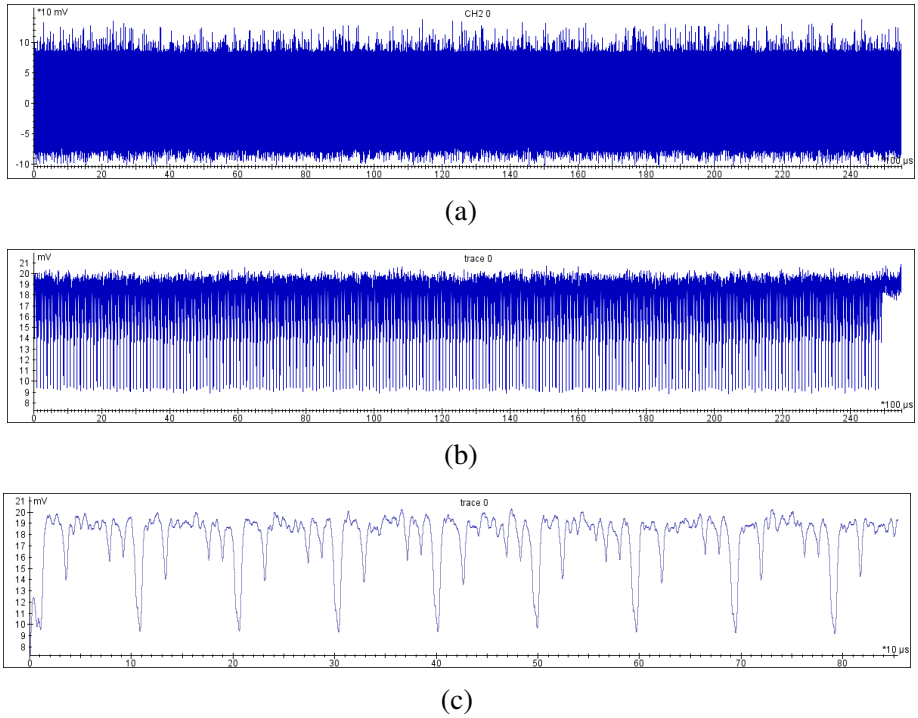


Figura 3.13: *Trace* de radiação eletromagnética de uma execução de ECSM com algoritmo Montgomery Ladder: (a) original não filtrado; (b) após aplicação de filtro baixa e (c) zoom no início deste último. *Traces* processados com Inspector [Riscure 2016].

em frequências indesejadas estejam presentes.

Segmentação. Os *traces* de potência medidos tipicamente correspondem à execução completa da operação criptográfica; sendo assim, são contíguos e contêm todas as rodadas (rounds) ou sub-operações executadas; no contexto da multiplicação escalar, as rodadas são as n iterações do laço do algoritmo de multiplicação escalar implementado. Tais *traces* devem ser primeiramente segmentados em iterações, de modo que um conjunto de n *subtraces* é obtido, cada um contendo as amostras correspondentes à respectiva iteração.

Tal segmentação pode ser realizada de diversas maneiras. Um método ingênuo é identificar os índices das amostras de início e fim da execução do laço da ECSM, e então dividir este segmento em n segmentos de igual (ou quase igual) comprimento, cada qual correspondendo a uma iteração. Tal método apresenta dois problemas: o primeiro é de que em geral é difícil identificar as amostras de início e fim; o segundo é que o comprimento dos segmentos das iterações podem variar devido ao clock *jitter*. Um método mais robusto, que ameniza tais dificuldades, é aplicar um filtro passa baixa forte, de modo a identificar segmentos do *trace* que se repetem de uma iteração para outra; localizar então picos nestes segmentos cuja distância entre si é aproximadamente a mesma (Tal distância é o valor aproximado do comprimento daquela iteração.) e por fim cortar o *trace* original utilizando tais comprimentos como referência. No entanto, devido às dificuldades anteriormente mencionadas, é provável que a segmentação obtida não seja perfeita, e portanto amostras do início de uma iteração poderão estar presentes no *trace* da iteração anterior,

bem como amostras do fim de uma iteração poderão estar no início do *trace* da iteração seguinte; algo análogo acontece com relação à primeira e à última iterações e as partes do *trace* externas ao laço da ECSM.

Alinhamento. Dado que a segmentação provavelmente não será perfeita, uma operação aritmética realizada no intervalo de amostras $tr_i[s..e]$ no *trace* da iteração i muito provavelmente ocorrerá em um itervalo diferente $tr_j[s'..e']$ no *trace* da iteração j. Logo, é necessário alinhar todos os *subtraces* de iterações da ECSM. A Section 3.6.6.1 mostra alguns *traces* resultantes da segmentação, antes e após alinhamento estático por correlação. Diversos algoritmos para alinhamento de *traces* para SCA são propostos na literatura, dentre eles o alinhamento estático e o alinhamento elástico [Woudenberg et al. 2011]. O alinhamento estático é um método simples que consiste, grosso modo, em escolher um *trace* de referência e deslizar incrementalmente o *trace* a ser alinhado sobre o de referência e computar a distância entre eles de acordo com alguma métrica, p.ex. o coeficiente de correlação de Pearson; após deslizar um dado número de posições dentro de uma janela, considera-se que o *trace* estará alinhado se for deslocado (*shifted*) até a posição cuja distância foi mínima.

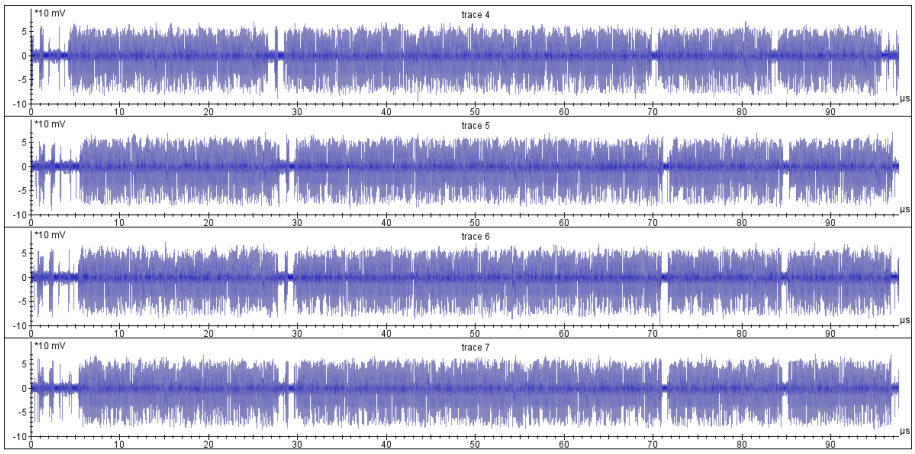
3.6.6.2. Algoritmos para *clustering*

Em SCA, métodos de análise baseados em aprendizado de máquina, tais como Support Vector Machines [Bartkewitz and Lemke-Rust 2013], Random Forests [Lerman et al. 2014], análise de séries temporais [Lerman et al. 2013] e análise nebulosa (*Fuzzy analysis*) [Saeedi and Kong 2014] tem sido recentemente empregados como uma alternativa ao método de *template attack* no contexto de ataques *profiled*, em particular quando a distribuição das amostras difere muito da distribuição gaussiana; e também no contexto de ataques não *profiled*, através da utilização de algoritmos de *clustering* não supervisionados.

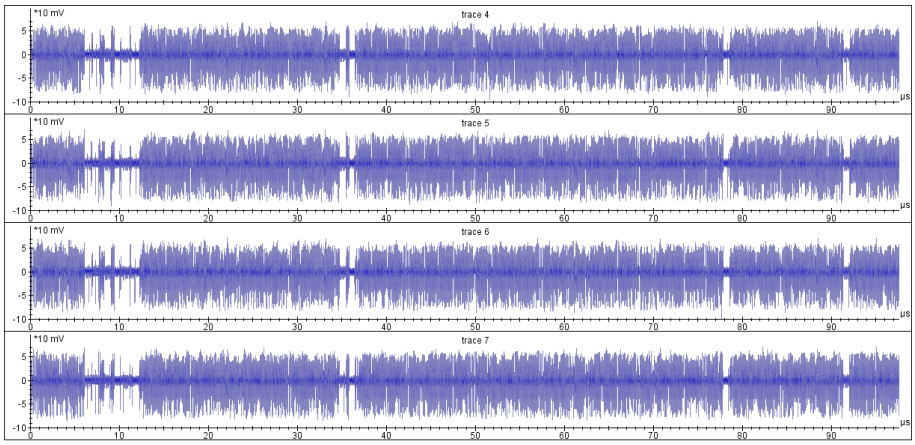
Dentre os algoritmos para *clustering* empregados com sucesso no contexto de ataques horizontais estão o K-Means [Forgy 1965, Lloyd 1982], Fuzzy K-Means [Dunn 1973] e o Expectation-Maximization (EM) [Dempster et al. 1977]. O K-Means é um algoritmo de *clustering* rígido, isto é, cada instância (uma amostra, no contexto de HCA) é atribuída (rotulada) a um único cluster. Fuzzy K-Means and EM, por outro lado, são algoritmos de *clustering* suaves (*soft*), pois tem como saída uma matriz de probabilidades de associação, onde a cada instância está associado o grau de vínculo desta com cada um dos clusters. Referimos o leitor aos livros sobre aprendizado de máquina [Alpaydin 2014, Witten and Frank 2011, Han et al. 2011, Bishop 2007, Duda et al. 2001] para descrições destes algoritmos e variantes.

3.6.6.3. Análise com chave conhecida

A análise com chave conhecida consiste em determinar os pontos de interesse, isto é, os índices de amostra, onde o vazamento é mais forte, com base no conhecimento da chave. Devido à necessidade de conhecimento do valor chave/escalar, tal análise é empregada somente na fase de teste do ataque HCA para determinar, por exemplo, quantos *traces* são

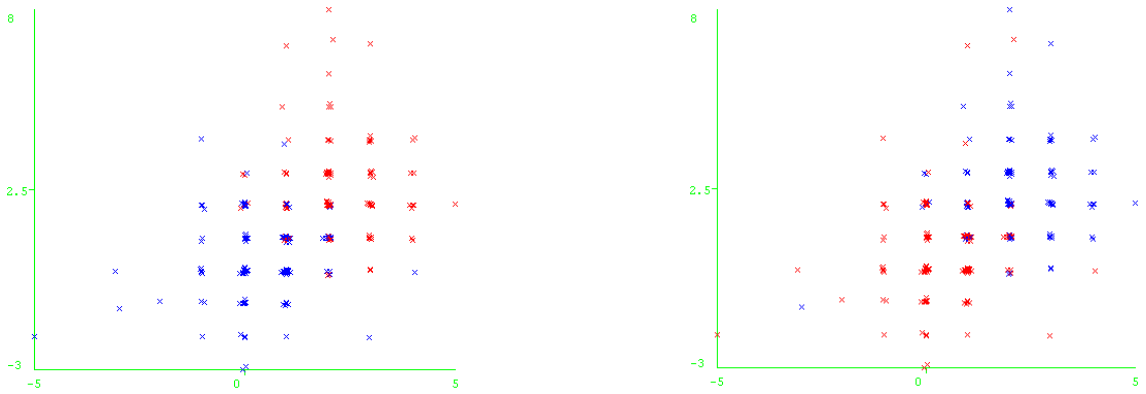


(a)



(b)

Figura 3.14: *Traces* EM de iterações da ECSM: (a) logo após segmentação, desalinhados; e (b) após alinhamento. *Traces* processados com Inspector [Riscure 2016].



(a) (b)

Figura 3.15: Visualização dos *clusters* obtidos pelo algoritmo EM em múltiplas dimensões/atributos. Cada dimensão corresponde ao índice de um POI, de um total de 20 POIs. Entretanto, para o propósito de visualização, apenas duas destas dimensões são exibidas. Em (a) estão os clusters obtidos pelo algoritmo; e em (b) os rótulos corretos. Amostras com a mesma cor foram rotuladas com o mesmo valor do bit.

necessários como entrada para a avaliação de vazamento de um dado dispositivo, de modo que os pontos de interesse obtidos por aquela correspondam aos previstos e sabidamente relevantes obtidos pela análise com chave conhecida.

Tal análise consiste em aplicar o algoritmo de clusterização no conjunto de amostras presentes em um dado índice de amostra (m), obtendo-se dois grupos de amostras: o primeiro grupo corresponde às amostras rotuladas com o valor b ($b \in \{0,1\}$, mas não se sabe se 0 ou 1) e o segundo grupo corresponde ao valor oposto $\bar{b}$ (Figure 3.15).

O conhecimento da chave é utilizado para determinar quantos bits tiveram o seu valor corretamente identificado no agrupamento. Como não se sabe o valor de b, isto é, o rótulo de cada grupo, o seguinte procedimento é adotado. Toma-se $b = 0$ e conta-se o número de bits corretamente identificados (n_c); o valor $n - n_c$ (n é o comprimento em bits do escalar) é então o número de bits corretamente identificados caso a rotulação esteja errada (isto é, o correto é $b = 1$). Finalmente, considera-se $max\{n_c, n - n_c\}$ o número de bits corretamente identificados.

O procedimento acima descrito é repetido para todos os índices de amostra, e o resultado são os pontos em que o vazamento de bits da chave é mais intenso. A ordenação destes pontos em ordem decrescente do número de bits corretamente identificados fornece os pontos de interesse.

3.6.6.4. Avaliação de vazamento

Técnicas de avaliação de vazamento determinam se um dispositivo criptográfico está vazando informação por canal lateral, com base em método estatístico e um modelo de va-

zamento. Em [Meynard et al. 2011] os autores testaram informação mútua (MIA)[4] como um método para localizar vazamento no domínio da frequência e, consequentemente, encontrar as bandas de frequência no *traces* de EM do RSA em que as diferenças entre quadrados e multiplicações são maiores. O Welch t-test é um outro método estatístico que pode ser empregado para este fim, p.ex., em metodologias como a TVLA (cf. Section 3.8.3.1). Os autores de [Mather et al. 2013] demonstraram como empregar t-test e MIA para localizar vazamento no domínio do tempo.

No escopo de ataques horizontais, tais métodos são empregados sem que haja conhecimento do valor da chave secreta ou de números aleatórios gerados e/ou usados pelo dispositivo. Portanto, são aplicáveis em um cenário realista onde o adversário não tem qualquer controle (escrita ou leitura) da chave do dispositivo ou outra informação secreta, em particular quando contramedidas como SR e CRR são aplicadas e não podem ser desabilitadas. Os pontos em que o vazamento é mais intenso, obtidos pela aplicação de um método para análise de vazamento, são tomados como pontos de interesse (POI). O valor das amostras em tais pontos são posteriormente utilizados na fase de ataque para recuperação da chave.

O método de análise de vazamento proposto em [Perin and Chmielewski 2015] demonstra como múltiplos *traces* podem ser combinados para a avaliação de vazamento, no contexto de ataques horizontais ao RSA. Tal método é baseado em *clustering* e funciona mesmo se o dispositivo empregar qualquer combinação das contramedidas clássicas aplicadas a implementações da exponenciação modular: exponent blinding, message or modulus randomization.[5]

3.6.6.5. Avaliação de vazamento baseada em *clustering*

Descrevemos nesta seção como o método de análise de vazamento de Perin e Chmielewski [Perin and Chmielewski 2015] pode ser adaptado a uma implementação do algoritmo Montgomery Ladder para multiplicação escalar. Supomos que deseja-se identificar o valor do bit do escalar utilizado em cada iteração do laço principal deste algoritmo através de algum vazamento direto ou indireto deste valor. Sejam n_0 e n_1 o número de bits 0 e 1 em um *trace*. A razão n_0/n_1 é aproximadamente constante, tendendo a 1, partindo da premissa de que os bits do escalar são gerados aleatoriamente. Devido à contramedida SR, o escalar efetivamente utilizado no laço da ECSM varia entre uma execução e outra da ECSM, e portanto difere de um *trace* para outro.

O método tem as seguintes premissas sobre o modelo de vazamento:

- **Premissa 1**: em um *trace* i, o valor médio para o conjunto de amostras em um índice m que correspondem às iterações cujo bit são 0 ou 1 são, respectivamente, $\mu_0^i + \gamma_0^i$ e $\mu_1^i + \gamma_1^i$, onde γ_k^i é um ruído aleatório com distribuição normal, $k = 0, 1$.
- **Premissa 2**: para todos os *traces* i, as médias μ_0^i e μ_1^i são constantes.

[4]Mutual Information Analysis.

[5]Exponent blinding e message randomization são equivalentes às contramedidas SR e CRR para ECC, respectivamente

Seja um *trace i* e as amostras localizadas em um índice *m* deste *trace*. A saída do algoritmo de *clustering* quando aplicado a este conjunto de amostras são dois centróides, $c_{0,m}$ e $c_{1,m}$ e dois clusters de amostras $\{g_{0,m}\}$ e $\{g_{1,m}\}$ contendo $p_{0,m}$ e $p_{1,m}$ elementos cada, respectivamente, tal que $p_{0,m} + p_{1,m} \approx n_0 + n_1$.

Temos então, para todo *trace i* e todo índice de amostra *m* deste *trace*, o seguinte conjunto de parâmetros: $c_{k,i}$, $\{g_{k,m}\}$, $p_{k,m} = |\{g_{k,m}\}|$ e $\sigma^2_{k,m} = Var(\{g_{k,m}\})$, para $k = 0, 1$. Este conjunto de parâmetros é utilizado como entrada para uma dentre as seguintes funções estatísticas, também conhecidas como *distinguishers*: diferença de médias (DoM), soma dos quadrados das diferenças (SOSD), soma dos quadrados dos t-values (SOST) e MIA.

Defina o seguintes parâmetros, para $k = 0, 1$:

$$r_{k,m} = \frac{min\{p_{k,m}, n_k\}}{max\{p_{k,m}, n_k\}}, \tag{4}$$

$$\beta_m = r_{0,m} \cdot r_{1,m}. \tag{5}$$

As funções distinguisher DoM, SOSD, SOST podem ser então definidas do seguinte modo[6]:

$$\begin{aligned}
\text{DoM} &: l_{\text{DOM},m} = |c_{0,m} - c_{1,m}| \\
\text{SOSD} &: l_{\text{SOSD},m} = |c_{0,m} - c_{1,m}|^2 \\
\text{SOST} &: l_{\text{SOST},m} = \left(\frac{|c_{0,m} - c_{1,m}|}{\sqrt{\frac{\sigma^2_{0,m}}{p_{0,m}} + \frac{\sigma^2_{1,m}}{p_{1,m}}}} \right)^2
\end{aligned}$$

A função *distinguisher* é aplicada em cada índice de amostra *m*, para cada *trace i*. O valor resultante é somado para todos os *traces*, a média é calculada e o valor é ajustado pelo coeficiente β_m, isto é, $\bar{l}_{\text{D},m} = \beta_m \frac{1}{N} \sum_{i=1}^{N} l^{(i)}_{\text{D},m}$, onde $D \in \{$ DoM,SOSD,SOST,MIA $\}$. O valor $\bar{l}_{\text{D},m}$ é, portanto, o valor estimado do vazamento no índice de amostra *m*, segundo a função *distinguisher D*.

A aplicação de algoritmos de *clustering* fornece uma estimativa para as médias $\mu_{k,m}$. Por causa do somatório usado na definição de $\bar{l}_{\text{D},m}$ e das premissas acima, o ruído $\gamma_{k,m}$ em cada amostra *m* é eliminado se o número de *traces* processados é suficientemente grande. A Figure 3.16 mostra o valor estimado do vazamento em cada índice de amostra para o *distinguisher* SOST aplicado a *traces* provenientes de uma implementação do algoritmo Montgomery Ladder.

[6]Referimos o leitor a [Perin and Chmielewski 2015] para a definição da função MIA neste caso.

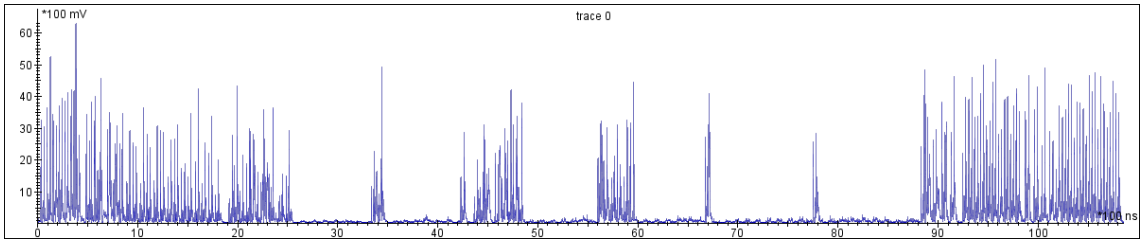


Figura 3.16: Estimativa de vazamento baseada em clusters utilizando o *distinguisher* SOST. Visualizado no Inspector [Riscure 2016].

3.6.6.6. Ataque para recuperação de chave

A etapa de ataque consiste na recuperação propriamente dita do escalar utilizando apenas um único *trace*, de uma única execução da ECSM. Esta etapa recebe como entradas o conjunto de *traces* e uma lista de n_{poi} POIs, e consiste nos seguintes passos:

Agrupamento. Para cada vetor de amostras em cada POI, é aplicado um dos algoritmos de *clustering* (K-Means, Fuzzy K-Means ou EM). O resultado são dois clusters para cada vetor de amostras, um correspondente ao bit 0 e o outro ao bit 1, e uma matriz de associação M_{nx2}, cujas entradas $m_{i,k}$ são a probabilidade de que a amostra i pertença à classe $k \in \{0,1\}$. Tal rotulamento das amostras pode ser visto como um escalar aproximado. A saída deste passo são n_{poi} escalares aproximados/candidatos, n_{poi} matrizes de associação.

Estimação do escalar final. Neste passo os escalares aproximados são combinados em um escalar final. Para tanto, um classificador estatístico (Majority Rule, Log-likelihood ou estimação de Bayes) é empregado.

Regra da Maioria. Este classificador simplesmente toma o rótulo da amostra de cada POI obtido pelo agrupamento (i.e., chute para o valor do bit) como um voto e então considera o valor que recebeu a maioria dos votos como o valor real do bit.

Log-likelihood e Estimação de Bayes. Estes são classificadores paramétricos; no contexto de SCA o modelo gaussiano é tipicamente adotado. Referimos o leitor a [Perin et al. 2014, Perin and Chmielewski 2015] para uma descrição do uso de tais classificadores no contexto de ataques horizontais baseados em *clustering*.

Cálculo do grau de confiança. Neste passo é calculado o grau de confiança (*confidence score*) no valor de cada bit recuperado. O grau de confiança é um número real entre 0 (total incerteza) e 1 (total certeza). O cálculo do grau de confiança depende de particularidades do ataque sendo realizado; exemplos: [Nascimento et al. 2016, Perin and Chmielewski 2015].

Cálculo da taxa de sucesso e nível de confiança. Se o valor do escalar é conhecido, isto é, se o ataque não está sendo aplicado na prática em um alvo real, mas sim sendo testado, então podem ser calculados também a taxa de sucesso e o nível de confiança do ataque.

A taxa de sucesso é simplesmente a média do número de bits do escalar que são corretamente recuperados quando o ataque é aplicado a um grande conjunto de *traces* de execuções completas da ECSM.

O nível de confiança é calculado da seguinte forma. Sejam C_{wrong} e C_{right} o conjunto de graus de confiança para os bits cujo valor recuperado está, respectivamente, errado ou certo, e $C_{\text{all}} = C_{\text{wrong}} \cup C_{\text{right}}$. Calcule $c_{\text{max,wrong}} = \max\{C_{\text{wrong}}\}$, $n_{\text{known\_wrong}} = |c \in C_{\text{all}}, c \leq c_{\text{max,wrong}}|$ e $n_{\text{known\_right}} = |C_{\text{all}}| - n_{\text{known\_wrong}}$. O nível de confiança é então definido por $\text{conf\_level} = n_{\text{known\_right}}/(n_{\text{known\_right}} + n_{\text{known\_wrong}})$ e representa a fração dos bits que foram corretamente recuperados com alta confiança, isto é, com confiança acima do limiar $c_{\text{max,wrong}}$. O nível de confiança indica a qualidade dos graus de confiança obtidos, isto é, quão bem eles permitem separar os bits do escalar cujo valor foi corretamente recuperado daqueles cujo valor recuperado está errado.

Ambos taxa de sucesso e nível de confiança são indicadores do sucesso de um ataque horizontal, e em última instância, se este é viável ou não, dadas as seguintes condições, dentre outras: qualidade e adequação do aparato de medição, SNR dos *traces* medidos, segmentação e alinhamento dos *traces*, qualidade dos pontos de interesse obtidos na etapa de avaliação de vazamento.

3.6.7. Ataques *template* versus Ataques horizontais

Precondições e limitações dos ataques baseados em *template*: Ataques baseados em *template* são os mais poderosos ataques do tipo SCA, segundo a Teoria da Informação [Chari et al. 2003]. No entanto, ataques baseados em *template* só podem ser realizados quando a contramedida SR não é aplicada ou quando pode ser desabilitada durante a fase de criação de *templates (profiling)*, caso contrário os *templates* não podem ser criados. Uma outra limitação deste tipo de ataque é de que dispositivos diferentes, mesmo que sejam do mesmo modelo, mesmo lote, etc., têm imperfeições únicas resultantes do processo de fabricação as quais resultam em diferenças no consumo de potência e radiação eletromagnética. Tais diferenças podem ser grandes o suficiente de modo que os *templates* gerados a partir dos *traces* provenientes do dispositivo de *profiling* não sejam bons modelos do vazamento observado no dispositivo alvo do ataque, assim reduzindo a taxa de sucesso do ataque [Elaabid and Guilley 2012].

Aplicabilidade. Até então estes ataques só foram demonstrados em CPUs embarcadas de 8, 16 e 32 bits, devido ao alto nível de SNR (*Signal-to-Noise Ratio*) que pode ser obtido na medição no consumo de potência e EM nestes dispositivos. Quando o SNR é baixo, além de haver pouco vazamento de dados (*data-leakage*) explorável do valor da chave ou valores intermediários derivados deste, o alinhamento dos *subtraces* torna-se também inviável, devido à inexistência de intervalos próximos da ocorrência da operação-alvo em que as amostras tem valores idênticos ou semelhantes em todos os *subtraces*.

3.7. Recuperação de chaves com erros em criptosistemas baseados no (EC)DLP

Devido ao ruído, vazamento de dado (por canal lateral) não relacionado à chave secreta, e outros aspectos que interferem com a análise por canal lateral (p.ex., desalinhamento,

clock jitter), o escalar final obtido por ataque SCA realizado a partir de um único *trace* provavelmente conterá erros, isto é, o valor de alguns dos bits recuperados estará incorreto.

Se a quantidade de bits incorretos (erros) é suficientemente pequena, então um ataque de força bruta pode ser viável, mesmo que não se saiba a localização de tais bits incorretos. A complexidade de tal busca, isto é, o número de escalares que devem ser testados até o escalar correto ser encontrado é $\binom{n}{s}2^s$, onde n é o comprimento do escalar em bits e s é número de erros. Considerando um escalar de $n = 256$ bits[7], o valor máximo de s para concluir tal busca em tempo aceitável é 6, o que significa que aproximadamente 2^{56} escalares precisam ser testados.

Se a quantidade de bits incorretos for maior, então o adversário necessita saber a localização dos possíveis bits incorretos no escalar recuperado para corrigi-los em tempo viável. Neste caso, a noção de bits suspeitos pode ser usada como referência para a seleção de bits do escalar com respeito a um ataque de força bruta. Um bit é considerado *suspeito* se o grau de confiança deste é menor do que o maior grau de confiança de qualquer bit falsamente/erroneamente identificado. Este último limiar (*threshold*) é determinado experimentalmente na fase de *profiling*.

Vamos supor que para um dado *trace* o escalar recuperado tenha $s = 54$ bits suspeitos, de um total de 254 bits. Para recuperar tal escalar, se este foi completamente aleatorizado pela contramedida SR, o adversário precisa realizar $O(2^{254})$ operações, no pior caso, o que geralmente não é prático.[8]

Para melhorar esta complexidade de força bruta, há duas opções. A primeira abordagem é tentar explorar a distribuição dos bits suspeitos nos conjuntos de bits recuperados incorretamente e corretamente (Figure 3.17). Enquanto há uma clara tendência dos bits incorretos terem um grau de confiança menor, a área da intersecção entre as duas distribuições é grande. Ainda assim, pode ser possível explorar esta tendência com um ataque de força bruta informado [Lange et al. 2015], priorizando os bits com os menores graus de confiança. Infelizmente esse ataque funciona bem se os bits contendo erros são adjacentes e este não é o caso no nosso contexto.

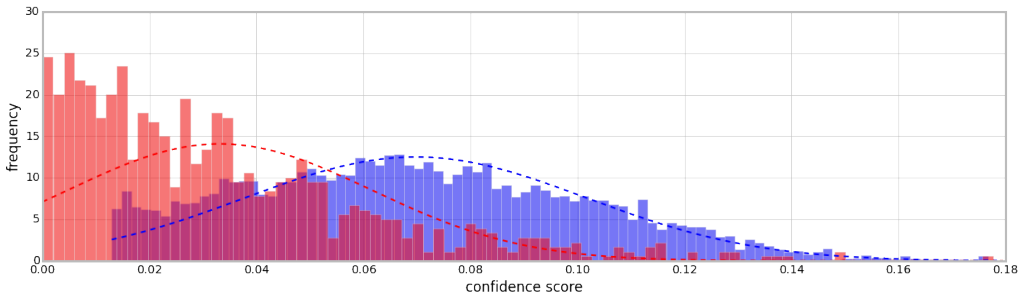


Figura 3.17: Distribuição de graus de confiança de bits suspeitos para um conjunto de 1000 *traces* de execuções completas da ECSM. Vermelho: bits recuperados incorretamente; azul: bits corretamente recuperados porém considerados suspeitos.

[7]comprimento típico do escalar para uma curva no nível de segurança de 128 bits

[8]Nesta subseção são consideradas apenas complexidades de pior caso.

3.7.1. Algoritmo de Gopalakrishnan, Theriault e Yao [Gopalakrishnan et al. 2007a]

Alternativamente, ou combinado com uma busca de força bruta informada, os autores de [Nascimento et al. 2016] aplicaram o segundo algoritmo de [Gopalakrishnan et al. 2007b] ao Montgomery Ladder, e este é o objeto desta seção. Tal algoritmo é baseado no paradigma de balanceamento entre tempo e memória (*time-memory trade-off*) e foi originalmente projetado para cadeias de cálculos de quadrados e multiplicação.

Descrevemos como o algoritmo funciona tomando como exemplo um escalar recuperado, contendo $s = 54$ bits suspeitos. Vamos representar os índices deste bits como uma lista ordenada em ordem decrescente: $i_s, \ldots i_1$, onde cada $i_j \in \{0, \ldots 254\}$ e $s \geq j \geq 1$; note que há 255 bits no total. Seja x o índice $i_{\lfloor \frac{s}{2}+1 \rfloor}$. Seja a o número representado pela cadeia de bits correspondente à parte esquerda do escalar (mais significativa) antes de x (incluindo i_x) e seja b o número correspondente à cadeia de bits da parte direita (menos significativa). Seja $y = 254 - i_x$ o comprimento da parte direita. Além disso, sabemos que $R = kP$, onde R é o ponto resultante, k é o escalar correto a ser recuperado e P é o ponto de entrada.

Então, claramente $R = kP = [a \cdot 2^{i_x} + b]P = [a]([2^{i_x}]P) + [b]P$. Se denotarmos $[2^{i_x}]P$ por H, então a equação acima se reduz a

$$R - [b]P = [a]H. \tag{6}$$

Podemos usar Equation (6) para verificar a corretude do chute. Seguindo [Gopalakrishnan et al. 2007b], usamos uma técnica de equilíbrio entre tempo e memória para acelerar uma busca exaustiva. Para cada "chute"do valor de a, computamos $[a]H$ e armazenamos todos os pares $(a, [a]H)$ em uma tabela. Então, ordenamos todos os pares pelo valor do ponto $A = [a]H$.

A seguir, "chutamos"um valor para b e computamos $Z = R - [b]P$. Se nosso chute para b estiver correto, então Z está presente na tabela, e o valor do escalar z correspondente (tal que $Z = [z]P$) é imediatamente obtido pois está na mesma linha desta tabela. Se Z está presente, então a busca termina e o escalar correto é $s^* = z||a$.

Como há aproximadamente $2^{\frac{s}{2}}$ possibilidades para a e b, a complexidade de tempo é $O(2^{\frac{s}{2}})$ operações. Como há $2^{\frac{s}{2}}$ possibilidades para a, a tabela tem aquele número de entradas e a complexidade de espaço é $O(2^{\frac{s}{2}})$. Desta forma nós limitamos a complexidade de tempo para $O(2^{\frac{s}{2}})$ (cf. [Gopalakrishnan et al. 2007b] para uma análise de complexidade detalhada), a qual é da ordem de 2^{54} para o escalar de exemplo.

Dado um conjunto de *traces* provenientes de medições da ECSM no dispositivo alvo, não sabemos qual *trace* contém o menor número de bits suspeitos pois não sabemos o grau de confiança máximo de um bit incorretamente identificado. No entanto, este valor pode ser determinado atacando alguns *traces* para os quais o escalar aleatorizado é conhecido.

No trabalho [Nascimento et al. 2016], os autores determinaram que, para o dispositivo e implementação em software da curva *Curve25519* considerados, o número 54 de bits suspeitos cobre todos os bits identificados incorretamente para pelo menos um

trace dentro de um conjunto de $m = 100$ *traces*. Logo, o ataque completo para recuperar o valor correto do escalar funciona do seguinte modo: o algoritmo acima descrito é executado para cada um dos *m traces*, em sequência; o ataque pára quando o algoritmo encontra um ponto na tabela, e portanto o escalar foi determinado; se nenhum dos pontos foi encontrado na tabela, para nenhum dos *traces*, então o ataque falhou.

Como o algoritmo é executado m vezes, a complexidade do ataque completo é multiplicada por m. A complexidade totaliza $O(n \cdot 2^{\frac{s}{2}})$ operações e $O(n \cdot 2^{\frac{s}{2}})$ em memória. Para o ataque da seção anterior, isso corresponde a $O\left(m \cdot 2^{27}\right) = O\left(2^{32}\right)$ operações.

3.8. Ferramentas

3.8.1. Ferramentas para verificação de tempo constante

Os primeiros métodos para verificação formal de contramedidas para canais laterais de tempo foram construídos a partir de análise estática. [Molnar et al. 2005] propõem métodos para detetar canais laterais de controle de fluxo e transformar código fonte em C para eliminar as vulnerabilidades, abrangendo ataques de tempo e tratamento de erros. [Coppens et al. 2009] modificam um compilador para converter comandos condicionais de forma que o código *Assembly* resultante não mais tenha comportamento no tempo dependente dos dados processados. [Lux and Starostin 2011] propõem uma ferramenta para detetar potenciais canais laterais em implementações em Java de algoritmos criptográficos, baseados em anotações do programador. Em [Köpf et al. 2012], os autores propõem um método novo baseado em limitantes superiores automaticamente derivados na quantidade de informação sobre a entrada que o adversário consegue extrair de um programa a partir da observação do comportamento da *cache* durante a execução. [Doychev et al. 2015] propõem métodos para calcular aproximações precisas da quantidade de informação vazada que podem ser observadas nos canais laterais a seguir: transições de estado na *cache*, traços de acertos e erros, e tempo de execução. Os autores também sugerem provas formais de segurança para contramedidas como pré-carga de endereços e padrão de acesso independente de dados.

Ainda considerando ataques de tempo, outras abordagens para verificação foram propostas recentemente. O trabalho [Almeida et al. 2013] considera as políticas em alto nível adotadas na implementação da biblioteca NaCl: ausência de desvios condicionais e endereçamento de vetores dependentes de dados; formalizam as políticas e propõem um método de verificação formal baseado em auto-composição, demonstrando-o pela aplicação no código *Assembly* otimizado de algumas funções da biblioteca. [Langley 2012] propõe um método de análise dinâmico baseado no módulo `memcheck` da ferramenta *Valgrind*, que amplifica sua capacidade para reconhecer dados não-inicializados na granularidade de bits. A ferramenta *FlowTracker*[9] [Rodrigues et al. 2016] foi proposta recentemente para verificar comportamento constante de código compilado pela análise estática de fluxo de informação na representação intermediária LLVM. As vantagens da ferramenta são a facilidade de descrição das interfaces e a baixa intrusão, já que nenhuma alteração é necessária no código. CT-Verif [Almeida et al. 2016] é uma ferramenta seguindo uma abordagem similar que fornece garantias formais adicionais, mas exige alteração do código pelo verifica-

[9] `http://cuda.dcc.ufmg.br/flowtracker/`

dor.

3.8.2. Ferramentas para verificação de implementações contra ataques de potência

Verificação formal de implementações em software de algoritmos criptográficos contra análise de potência é um assunto que tem sido pesquisado recentemente. Por exemplo, Maggi et al. [MAGGI 2013, Agosta et al. 2013] propuseram um método baseado em análise de fluxo de dados para identificar dependências entre as instruções executadas e dados secretos. O método é implementado em um compilador LLVM como uma passada especializada operando no nível de representação intermediária, portanto ela é agnóstica em arquitetura, suportando quaisquer das arquiteturas de computador suportadas por LLVM. A ferramenta automaticamente instancia contramedidas de masking à implementação.

Mais recentemente, Bayrak et al [Bayrak et al. 2013, Bayrak et al. 2014] mostraram como reduzir o problema de verificação da resistência de uma implementação a vazamento por canais de potência a um conjunto de problemas SAT, os quais podem ser eficientemente tratados pelos resolvedores SAT atuais. Tal método, em princípio, trata das limitações da abordagem baseada em análise de fluxo de informação.

Resolvedores baseados em teorias do módulo de satisfabilidade (SMT) foram aplicados por Eldib et al [Eldib and Wang 2014, Eldib et al. 2014b, Eldib et al. 2014a, Eldib et al. 2014c] a este problema. Estes últimos também propuseram métodos para a aplicação automatizada de contramedidas.

3.8.3. Métodos empíricos para análise de vazamentos

Avaliações de segurança de dispositivos criptográficos com respeito a canais laterais compreende duas fases: *medição* e *análise*. A saída ou resultado de tal avaliação deve ser Falhou (*Fail*) ou Passou (*Pass*). O resultado de tal avaliação deve ser interpretado segundo as restrições do processo de avaliação, tais como a acurácia do equipamento de teste, expertise técnica dos avaliadores e tempo disponível para a avaliação. A medição dos *traces* e suas limitações deve ser levada em consideração, caso contrário a fase posterior, de análise, pode ser prejudicada ou invalidada, resultando em falsos positivos, ou pior, em falsos negativos.

As metodologias de avaliação atuais (p.ex., Common Criteria [Criteria 2014]) consistem na realização de uma bateria de ataques por canais laterais conhecidos contra o dispositivo sob teste (DUT) [10], numa tentativa de recuperar a chave. Apesar disto, a rápida evolução das técnicas de análise por canais laterais atuais propostas na literatura tem exigido um nível crescente de expertise dos testadores e um aumento no tempo requerido para avaliação. Mesmo quando todas as tentativas de ataque falham, vazamentos residuais no canal lateral avaliado podem ainda estar presentes, o que pode revelar novos caminhos de ataque (*attack paths*) para um adversário.

[10] Device under test.

3.8.3.1. Test Vector Leakage Assessment (TVLA)

Por estas razões o NIST organizou um workshop em 2011 [NIST 2011] para encorajar o desenvolvimento de métodos de teste, métricas e ferramentas para avaliação da eficácia de mitigações contra ataques não-invasivos a módulos criptográficos.

Nesse workshop, a CRI[11] propôs a metodologia *Test Vector Leakage Assessment* (TVLA) [Goodwill et al. 2011] com o propósito de resolver os problemas acima. Os autores dessa metodologia consideram que duas figuras de mérito são importantes: a eficácia, no sentido de que ela é reprodutível e é um indicador confiável da resistência atingida pelo dispositivo; e a relação custo-benefício, isto é, na palavra dos autores, "validar um nível moderado de resistência (p.ex., FIPS 140 nível 3 ou 4) não deve requerer uma quantidade excessiva de tempo de teste por algoritmo ou de habilidade do operador de teste" [Goodwill et al. 2011]. A abordagem da metodologia TVLA difere fundamentalmente das estratégias de avaliação focadas em ataque atualmente empregadas, adotando uma estratégia caixa-preta com foco na detecção de vazamento.

A fase de medição na TVLA é baseada na aquisição de *trace* de canal lateral quando vetores de teste padronizados são fornecidos como entrada para a implementação sob teste, e estabelece requisitos para os equipamentos e setup de medição, alinhamento e pré-processamento dos *traces*.

A fase de análise compreende teste de hipótese estatístico, mais especificamente, o t-test de Welch, o qual é capaz de detetar diferentes tipos de vazamento e permite ao analista identificar pontos no tempo que merecem investigação adicional.

A metodologia TVLA até então foi aplicada aimplementações do AES em hardware e software [Goodwill et al. 2011, Cooper et al. 2013, Mather et al. 2013], e implementações em software do RSA [Witteman et al. 2011a] e ECC (multiplicação escalar com base variável) [Nascimento et al. 2015]. Bem recentemente, os autores da TVLA detalharam mais a aplicação da metodologia ao RSA e como adaptá-la aos esquemas ECDSA e ECDH. [Tunstall and Goodwill 2016].

3.8.3.2. Outras metodologias para análise de vazamentos

Outras metodologias, baseadas em informação mútua contínua [Chothia and Guha 2011] e discreta [Chatzikokolakis et al. 2010] também foram propostas. Oswald et al [Mather et al. 2013] analisaram as metodologias [Goodwill et al. 2011], [Chatzikokolakis et al. 2010] e [Chothia and Guha 2011], e concluiu que elas têm poder estatístico similar. O trabalho recente de Schneider e Moradi [Schneider and Moradi 2016] aborda como realizar o teste t (em [Goodwill et al. 2011]) em ordens mais altas e como estendê-lo para o contexto multivariado.

[11] Empresa "Cryptography Research", atualmente incorporada à Rambus.

3.8.3.3. Aplicação da metodologia TVLA para ECC

Nesta subseção mostramos a aplicação da metodologia TVLA a uma implementação para microcontrolador AVR do esquema ECDH utilizando a curva *Curve25519* e o algoritmo Montgomery Ladder para ECSM protegida com a contramedida de aleatorização de coordenadas projetivas (CR) [Nascimento et al. 2015]. Especificamente, os conjuntos de vetores de teste na Table 3.4) foram selecionados para serem usados para a fase de medição de consumo de potência, os quais cobrem casos normais e especiais da aritmética de corpo finito e de grupo. A Table 3.5 mostra categorias de valores especiais usados nos conjuntos 4 e 5 para a função *compute shared secret* do esquema ECDH-Curve25519.

Tabela 3.4: Conjuntos de vetores de teste para análise de vazamento SPA (k é um escalar secreto e P é um ponto).

# Conj.	Propriedades	Descrição
1	k constante, P constante	Este é o baseline. Os testes comparam os *traces* dos outros conjuntos contra este.
2	k constante, P varia	Meta é detetar relações sistemáticas entre consumo de potência e o valor de P.
3	k varia, P constante	Meta é detetar relações sistemáticas entre consumo de potência e o valor de k.
4	k constante, P especial	Casos de borda dos algoritmos utilizados.
5	k especial, P constante	Casos de borda dos algoritmos utilizados.

Tabela 3.5: Categorias de valores especiais para n e q na função *compute shared secret* em ECDH-Curve25519 (q é um ponto codificado, n é um escalar codificado e l é a ordem do subgrupo).

Cat. #	Propriedades
1	$q \in \{0, 1, ..., 1023\}$
2	$q \in \{p_{25519} - 1, ..., p_{25519} - 1024\}$
3	$n \in \{0, ..., 1023\}$
4	$n \in \{l - 1, ..., l - 1024\}$
5	q tem um alto peso de Hamming (≥ 230)
6	q tem um baixo peso de Hamming (≤ 25)

Fase de aquisição. Os autores [Nascimento et al. 2015] capturaram 200 *traces* de potência para cada um dos conjuntos de vetores de teste, totalizando 1000 *traces*. A fase de análise de vazamento é idêntica à proposta na metodologia TVLA para o RSA [Witteman et al. 2011a], e é conduzida da seguinte forma. Sejam $\{DS_1, \ldots, DS_5\}$ os conjuntos de *traces* de potência correspondentes aos conjuntos de vetores de teste selecionados.

O teste completo consiste em executar os testes pareados descritos em [Witteman et al. 2011a] para cada um dos seguintes pares de datasets: $\{(DS_1, DS_2)$, $\ldots$, $(DS_1, DS_5)\}$. Se qualquer um dos testes anteriores falha, então o resultado da avaliação da implementação é *FALHOU*. Caso contrário, o resultado é *PASSOU*. Nós escolhemos o limiar de confiança $C = 4.5$, o mesmo valor usado na metodologia TVLA para o RSA [Witteman et al. 2011a].

Fase de análise SPA. A Figure 3.18 mostra a estatística t para um pequeno intervalo de índices de amostra [12] (i.e., instantes de tempo), para uma execução do teste t para grupo A $(S_{A,1}, S_{A,2})$ de vetores selecionados de DS_1 e DS_3, e o mesmo teste executado sobre o grupo (independente) B $(S_{B,1}, S_{B,2})$. [13]

A estatística t para o grupo A está acima do limiar $C = 4.5$ em um instante de tempo, significando uma possível dependência forte entre o consumo de potência e o valor da chave naquele instante. Mas, como este evento não ocorreu ao mesmo tempo e na mesma direção para o grupo B, ele é considerado um falso positivo pela metodologia e assim é descartado.

Os resultados de teste para cada par de conjuntos de vetores de teste $\{(DS_1, DS_2), \ldots, (DS_1, DS_5)\}$ mostrou que em poucos instantes de tempo o valor da estatística t para um dos grupos esteve acima de 4.5 ou abaixo de -4.5, mas nunca para ambos os grupos ao mesmo tempo. Portanto, pode-se concluir que a implementação passou a avaliação de vazamento SPA segundo a metodologia TVLA.

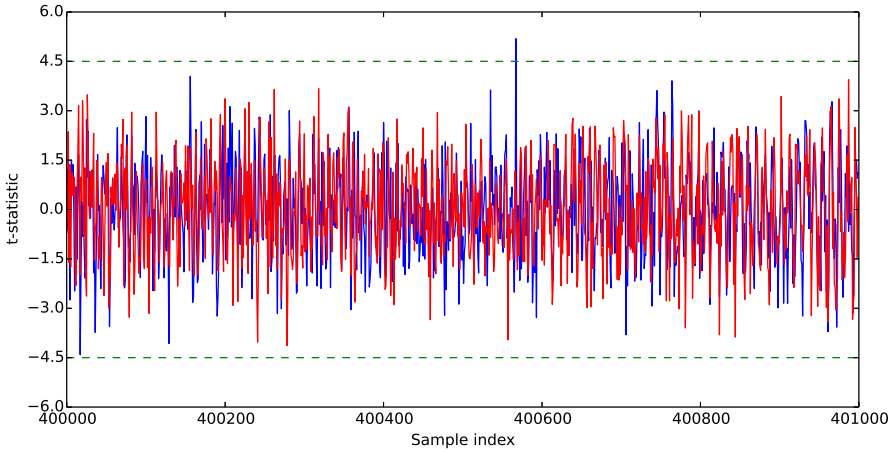


Figura 3.18: t-estatística versus índice de amostra para o experimento comparando DS_1 e DS_3, para dois grupos de *traces* independentes; grupo A (azul) e grupo B (vermelho).

3.8.4. Ferramentas para recuperação de chaves com erros em criptossistemas baseados no (EC)DLP

Lange et al [Lange et al. 2015] propuseram um algoritmo chamado ε-*enumeration* para computação do *rank* de uma chave usando como entrada as probabilidades/graus de confiança obtidos do ataque SCA juntamente com uma variação do algoritmo kangaroo de Pollard. O código fonte da implementação deste algoritmo foi disponibilizada pelos autores [Vredendaal 2014].

Os autores de [Nascimento et al. 2016] disponibilizaram o código fonte da implementação do algoritmo de Gopalakrishnan et al [Gopalakrishnan et al. 2007b]

[12] Este intervalo de tempo foi selecionado porque ele ilustra um intervalo onde os valores da estatística t são altos comparados com os os instantes de tempo

[13] Os grupos A e B são uma partição dos conjuntos de vetores de teste DS_1 e DS_3: $(S_{A,1} \subset DS_1,\ \ S_{A,2} \subset DS_3)$ e $(S_{B,1} = DS_1 \setminus S_{A,1},\ \ S_{B,2} = DS_3 \setminus S_{A,2})$.

(cf. Section 3.7.1) para recuperação de chaves com erros [Nascimento 2016]. Tal implementação foi otimizada para ECSM por Montgomery Ladder na Curve25519.

Referências

[AES 2001] (2001). FIPS 197 - Advanced Encryption Standard (AES). Technical report, National Institute of Standards and Technology.

[SHA 2012] (2012). FIPS 180-4 - Secure Hash Standard (SHA). Technical report, National Institute of Standards and Technology.

[Aciiçmez et al. 2007] Aciiçmez, O., Koç, c. K., and Seifert, J.-P. (2007). On the power of simple branch prediction analysis. In *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 312–320, New York, NY, USA. ACM.

[Agosta et al. 2013] Agosta, G., Barenghi, A., Maggi, M., and Pelosi, G. (2013). Compiler-based Side Channel Vulnerability Analysis and Optimized Countermeasures Application. In *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, pages 81:1—-81:6, New York, NY, USA. ACM.

[Akishita and Takagi 2003] Akishita, T. and Takagi, T. (2003). Zero-Value Point Attacks on Elliptic Curve Cryptosystem. pages 218–233.

[AlFardan and Paterson 2013] AlFardan, N. J. and Paterson, K. G. (2013). Lucky thirteen: Breaking the TLS and DTLS record protocols. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 526–540. IEEE Computer Society.

[Almeida et al. 2016] Almeida, J. B., Barbosa, M., Barthe, G., Dupressoir, F., and Emmi, M. (2016). Verifying constant-time implementations. In Holz, T. and Savage, S., editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 53–70. USENIX Association.

[Almeida et al. 2013] Almeida, J. B., Barbosa, M., Pinto, J. S., and Vieira, B. (2013). Formal verification of side-channel countermeasures using self-composition. *Sci. Comput. Program.*, 78(7):796–812.

[Alpaydin 2014] Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

[Bartkewitz and Lemke-Rust 2013] Bartkewitz, T. and Lemke-Rust, K. (2013). *Efficient Template Attacks Based on Probabilistic Multi-class Support Vector Machines*, pages 263–276. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Batina et al. 2014] Batina, L., Chmielewski, L., Papachristodoulou, L., Schwabe, P., and Tunstall, M. (2014). Online Template Attacks. In *Progress in Cryptology – INDOCRYPT 2014*, volume 1977, pages 21–36.

[Bauer and Jaulmes 2013] Bauer, A. and Jaulmes, É. (2013). Correlation analysis against protected SFM implementations of RSA. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8250 LNCS:98–115.

[Bauer et al. 2013] Bauer, A., Jaulmes, É., Prouff, E., and Wild, J. (2013). Horizontal and Vertical Side-Channel Attacks against Secure {RSA} Implementations. In *CT-RSA*, pages 1–17.

[Bayrak et al. 2014] Bayrak, A., Regazzoni, F., Novo Bruna, D., Brisk, P., Standaert, F., and Ienne, P. (2014). Automatic Application of Power Analysis Countermeasures. *Computers, IEEE Transactions on*, PP(99):1.

[Bayrak et al. 2013] Bayrak, A. G., Regazzoni, F., Novo, D., and Ienne, P. (2013). Sleuth: automated verification of software power analysis countermeasures. In *Cryptographic Hardware and Embedded Systems-CHES 2013*, pages 293–310. Springer.

[Bernstein 2004] Bernstein, D. J. (2004). Cache-timing attacks on AES. URL: `http://cr.yp.to/papers.html#cachetiming`.

[Bernstein 2005] Bernstein, D. J. (2005). Chacha20, a variant of salsa20. `https://cr.yp.to/chacha/chacha-20080120.pdf`.

[Bernstein et al. 2012] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., and Yang, B.-Y. (2012). High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89.

[Bertoni et al. 2008] Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2008). On the indifferentiability of the sponge construction. In Smart, N. P., editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer. `http://sponge.noekeon.org/`.

[Biham 1997] Biham, E. (1997). A fast new DES implementation in software. In Biham, E., editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 260–272. Springer.

[Bishop 2007] Bishop, C. (2007). *Pattern Recognition and Machine Learning*. Springer.

[Bonneau and Mironov 2006] Bonneau, J. and Mironov, I. (2006). Cache-Collision Timing Attacks Against AES. In Goubin, L. and Matsui, M., editors, *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*, pages 201–215. Springer.

[Brown et al. 2001] Brown, M., Hankerson, D., López, J., and Menezes, A. (2001). *Software Implementation of the NIST Elliptic Curves Over Prime Fields*, pages 250–265. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Brumley and Boneh 2003] Brumley, D. and Boneh, D. (2003). Remote timing attacks are practical. In *SSYM'03: Proceedings of the 12th conference on USENIX Security Symposium*, pages 1–1, Berkeley, CA, USA. USENIX Association.

[Chari et al. 2003] Chari, S., Rao, J. R., and Rohatgi, P. (2003). Template Attacks. *CHES 2002*, 2523:13–28.

[Chatzikokolakis et al. 2010] Chatzikokolakis, K., Chothia, T., and Guha, A. (2010). Statistical Measurement of Information Leakage. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *LNCS*, pages 390–404. Springer.

[Chothia and Guha 2011] Chothia, T. and Guha, A. (2011). A statistical test for information leaks using continuous mutual information. *Proceedings - IEEE Computer Security Foundations Symposium*, pages 177–190.

[Ciet and Joye 2003] Ciet, M. and Joye, M. (2003). ({Virtually}) Free Randomization Techniques for Elliptic Curve Cryptography. In *Information and Communications Security*, pages 348–359.

[Clavier et al. 2012] Clavier, C., Feix, B., Gagnerot, G., Giraud, C., Roussellet, M., and Verneuil, V. (2012). {ROSETTA} for Single Trace Analysis. pages 140–155.

[Clavier et al. 2010] Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., and Verneuil, V. (2010). *Horizontal Correlation Analysis on Exponentiation*, pages 46–61. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Clavier and Joye 2001] Clavier, C. and Joye, M. (2001). Universal Exponentiation Algorithm - A First Step towards Provable SPA-Resistance. In Koç, Ç., Naccache, D., and Paar, C., editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 300–308. Springer Berlin / Heidelberg.

[Cooper et al. 2013] Cooper, J., Demulder, E., Goodwill, G., Jaffe, J., and Kenworthy, G. (2013). Test Vector Leakage Assessment (TVLA) methodology in practice (Extended Abstract). Technical report, Cryptography Research Inc.

[Coppens et al. 2009] Coppens, B., Verbauwhede, I., Bosschere, K. D., and Sutter, B. D. (2009). Practical mitigations for timing-based side-channel attacks on modern x86 processors. In *30th IEEE Symposium on Security and Privacy (S&P 2009), 17-20 May 2009, Oakland, California, USA*, pages 45–60. IEEE Computer Society.

[Coron 1999] Coron, J.-S. (1999). Resistance against differential power analysis for elliptic curve cryptosystems. In *Cryptographic Hardware and Embedded Systems*, pages 292–302. Springer.

[Costello and Longa 2015] Costello, C. and Longa, P. (2015). Fourq: four-dimensional decompositions on a q-curve over the mersenne prime. *IACR Cryptology ePrint Archive*, 2015:565.

[Criteria 2014] Criteria, C. (2014). Common Criteria v3.1. Technical report, Common Criteria.

[Damgård 1989] Damgård, I. (1989). A design principle for hash functions. In Brassard, G., editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer.

[Danger et al. 2013] Danger, J.-L., Guilley, S., Hoogvorst, P., Murdica, C., and Naccache, D. (2013). A synthesis of side-channel attacks on elliptic curve cryptography in smart-cards. *Journal of Cryptographic Engineering*, 3(4):241–265.

[Dempster et al. 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

[Denis 2006] Denis, T. S. (2006). *BigNum Math: Implementing Cryptographic Multiple Precision Arithmetic*. Syngress Publishing.

[Dinur and Shamir 2012] Dinur, I. and Shamir, A. (2012). Applying cube attacks to stream ciphers in realistic scenarios. *Cryptography and Communications*, 4(3-4):217–232.

[Doychev et al. 2015] Doychev, G., Köpf, B., Mauborgne, L., and Reineke, J. (2015). Cacheaudit: A tool for the static analysis of cache side channels. *ACM Trans. Inf. Syst. Secur.*, 18(1):4.

[Duda et al. 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons.

[Düll et al. 2015] Düll, M., Haase, B., Hinterwälder, G., Hutter, M., Paar, C., Sánchez, A. H., and Schwabe, P. (2015). High-speed curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. *Des. Codes Cryptography*, 77(2-3):493–514.

[Dunn 1973] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

[Elaabid and Guilley 2012] Elaabid, M. and Guilley, S. (2012). Portability of templates. *Journal of Cryptographic Engineering*, pages 63–74.

[Eldib and Wang 2014] Eldib, H. and Wang, C. (2014). Synthesis of Masking Countermeasures against Side Channel Attacks. In Biere, A. and Bloem, R., editors, *Computer Aided Verification SE - 8*, volume 8559 of *Lecture Notes in Computer Science*, pages 114–130. Springer International Publishing.

[Eldib et al. 2014a] Eldib, H., Wang, C., and Schaumont, P. (2014a). SMT-Based Verification of Software Countermeasures against Side-Channel Attacks. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 62–77. Springer.

[Eldib et al. 2014b] Eldib, H., Wang, C., Taha, M., and Schaumont, P. (2014b). QMS: Evaluating the Side-Channel Resistance of Masked Software from Source Code. In *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*, pages 1–6. ACM.

[Eldib et al. 2014c] Eldib, H., Wang, C., Taha, M., and Schaumont, P. (2014c). SC Sniffer - Side-channel leak sniffer.

[Forgy 1965] Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769.

[Fouque and Valette 2003] Fouque, P. and Valette, F. (2003). The doubling attack - *Why Upwards Is Better than Downwards*. In Walter, C. D., Koç, Ç. K., and Paar, C., editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 269–280. Springer.

[Gierlichs et al. 2009] Gierlichs, B., Batina, L., Preneel, B., and Verbauwhede, I. (2009). Revisiting higher-order DPA attacks: Multivariate mutual information analysis. *IACR Cryptology ePrint Archive*, 2009:228.

[Goodwill et al. 2011] Goodwill, G., Jun, B., Jaffe, J., and Rohatgi, P. (2011). A testing methodology for side channel resistance validation. Technical report, CRI.

[Gopalakrishnan et al. 2007a] Gopalakrishnan, K., Thériault, N., and Yao, C. Z. (2007a). Solving Discrete Logarithms from Partial Knowledge of the Key. In *INDOCRYPT*, pages 224–237.

[Gopalakrishnan et al. 2007b] Gopalakrishnan, K., Thériault, N., and Yao, C. Z. (2007b). Solving discrete logarithms from partial knowledge of the key. In K. Srinathan, C. Pandu Rangan, M. Y., editor, *Progress in Cryptology – INDOCRYPT 2007*, volume 4859 of *LNCS*, pages 224–237. Springer.

[Goubin 2003] Goubin, L. (2003). A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. pages 199–210.

[Hamburg 2009] Hamburg, M. (2009). Accelerating AES with Vector Permute Instructions. In Clavier, C. and Gaj, K., editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 18–32. Springer.

[Han et al. 2011] Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

[Hankerson et al. 2003] Hankerson, D., Menezes, A. J., and Vanstone, S. (2003). *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Hankerson et al. 2004] Hankerson, D., Vanstone, S., and Menezes, A. J. (2004). *Guide to elliptic curve cryptography*. Springer.

[Hennessy and Patterson 2002] Hennessy, J. L. and Patterson, D. A. (2002). *Computer Architecture: A Quantitative Approach (The Morgan Kaufmann Series in Computer Architecture and Design)*. Morgan Kaufmann.

[Heyszl et al. 2014] Heyszl, J., Ibing, A., Mangard, S., Santis, F., and Sigl, G. (2014). Clustering Algorithms for Non-profiled Single-Execution Attacks on Exponentiations. In Francillon, A. and Rohatgi, P., editors, *CARDIS*, pages 79–93, Cham. Springer International Publishing.

[Ishai et al. 2003] Ishai, Y., Sahai, A., and Wagner, D. (2003). Private circuits: Securing hardware against probing attacks. In Boneh, D., editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer.

[Jean-Pierre et al. 2006] Jean-Pierre, O. A., pierre Seifert, J., and Çetin Kaya Koç (2006). Predicting secret keys via branch prediction. In *in Cryptology – CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007*, pages 225–242. Springer-Verlag.

[Jr. et al. 2016] Jr., M. A. S., Silva, M. V., Alves, R. C., and Shibata, T. K. (2016). Lightweight and escrow-less authenticated key agreement for the internet of things. *Computer Communications*, pages –.

[Käsper and Schwabe 2009] Käsper, E. and Schwabe, P. (2009). Faster and timing-attack resistant AES-GCM. In Clavier, C. and Gaj, K., editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 1–17. Springer.

[Koblitz 1987] Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209.

[Kocher 1996] Kocher, P. C. (1996). Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Koblitz, N., editor, *16th Annual International Cryptology Conference (CRYPTO 1996)*, volume 1109 of *LNCS*, pages 104–113. Springer.

[Kocher et al. 1999] Kocher, P. C., Jaffe, J., and Jun, B. (1999). Differential power analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, London, UK, UK. Springer-Verlag.

[Köpf et al. 2012] Köpf, B., Mauborgne, L., and Ochoa, M. (2012). Automatic quantification of cache side-channels. In Madhusudan, P. and Seshia, S. A., editors, *Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings*, volume 7358 of *Lecture Notes in Computer Science*, pages 564–580. Springer.

[Lange et al. 2015] Lange, T., van Vredendaal, C., and Wakker, M. (2015). Kangaroos in side-channel attacks. In Joye, M. and Moradi, A., editors, *Smart Card Research and Advanced Applications*, volume 8968 of *LNCS*, pages 104–121. Springer.

[Langley 2012] Langley, A. (2012). Ctgrind: Checking that functions are constant time with Valgrind. `https://github.com/agl/ctgrind`.

[Lerman et al. 2013] Lerman, L., Bontempi, G., Ben Taieb, S., and Markowitch, O. (2013). *A Time Series Approach for Profiling Attack*, pages 75–94. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Lerman et al. 2014] Lerman, L., Bontempi, G., and Markowitch, O. (2014). Power analysis attack: an approach based on machine learning. *International Journal of Applied Cryptography*, 3(2):97–115.

[Lloyd 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

[Luby and Rackoff 1988] Luby, M. and Rackoff, C. (1988). How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386.

[Lux and Starostin 2011] Lux, A. and Starostin, A. (2011). A tool for static detection of timing channels in java. *J. Cryptographic Engineering*, 1(4):303–313.

[MAGGI 2013] MAGGI, M. (2013). Automated side channel vulnerability detection and countermeasure application via compiler based techniques.

[Mather et al. 2013] Mather, L., Oswald, E., Bandenburg, J., and Wójcik, M. (2013). Does My Device Leak Information? An a priori Statistical Power Analysis of Leakage Detection Tests. In *Advances in Cryptology-ASIACRYPT 2013*, pages 486–505. Springer.

[McEvoy et al. 2007] McEvoy, R. P., Tunstall, M., Murphy, C. C., and Marnane, W. P. (2007). Differential power analysis of HMAC based on sha-2, and countermeasures. In Kim, S., Yung, M., and Lee, H., editors, *Information Security Applications, 8th International Workshop, WISA 2007, Jeju Island, Korea, August 27-29, 2007, Revised Selected Papers*, volume 4867 of *Lecture Notes in Computer Science*, pages 317–332. Springer.

[McGrew and Viega 2004] McGrew, D. A. and Viega, J. (2004). The security and performance of the galois/counter mode (GCM) of operation. In Canteaut, A. and Viswanathan, K., editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer.

[Merkle 1979] Merkle, R. C. (1979). *Secrecy, authentication, and public key systems.* PhD thesis, Stanford University.

[Meynard et al. 2011] Meynard, O., Réal, D., Flament, F., Guilley, S., Homma, N., and Danger, J. L. (2011). Enhancement of simple electro-magnetic attacks by pre-characterization in frequency domain and demodulation techniques. In *2011 Design, Automation Test in Europe*, pages 1–6.

[Miller 1986] Miller, V. S. (1986). Use of elliptic curves in cryptography. In Williams, H. C., editor, *Proceedings of CRYPTO 85*, pages 417–426. Springer. Lecture Notes in Computer Science No. 218.

[Molnar et al. 2005] Molnar, D., Piotrowski, M., Schultz, D., and Wagner, D. (2005). The program counter security model: Automatic detection and removal of control-flow side channel attacks. In Won, D. and Kim, S., editors, *Information Security and Cryptology - ICISC 2005, 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers*, volume 3935 of *Lecture Notes in Computer Science*, pages 156–168. Springer.

[Murdica et al. 2012] Murdica, C., Guilley, S., Danger, J.-L., Hoogvorst, P., and Naccache, D. (2012). Same Values Power Analysis Using Special Points on Elliptic Curves. In Schindler, W. and Huss, S., editors, *Constructive Side-Channel Analysis and Secure Design*, volume 7275 of *Lecture Notes in Computer Science*, pages 183–198. Springer Berlin / Heidelberg.

[Nascimento 2016] Nascimento, E. (2016). SAC 2016 - Implementation of algorithm for ECDLP with errors based on a time-memory tradeoff. `https://github.com/enascimento/SCA-ECC-keyrecovery`.

[Nascimento et al. 2016] Nascimento, E., Chmielewski, L., Oswald, D., and Schwabe, P. (2016). Attacking embedded ecc implementations through cmov side channels. In *23rd Conference on Selected Areas in Cryptography (SAC 2016), St John's, Canada, August 10-12, 2016.*

[Nascimento et al. 2015] Nascimento, E., López, J., and Dahab, R. (2015). Efficient and secure elliptic curve cryptography for 8-bit avr microcontrollers. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 289–309. Springer.

[NIST 2011] NIST (2011). Non-Invasive Attack Testing Workshop.

[Okeya 2006] Okeya, K. (2006). Side channel attacks against hmacs based on block-cipher based hash functions. In Batten, L. M. and Safavi-Naini, R., editors, *Information Security and Privacy, 11th Australasian Conference, ACISP 2006, Melbourne, Australia, July 3-5, 2006, Proceedings*, volume 4058 of *Lecture Notes in Computer Science*, pages 432–443. Springer.

[Percival 2005] Percival, C. (2005). Cache missing for fun and profit. In *Proceedings of BSDCan 2005*.

[Perin and Chmielewski 2015] Perin, G. and Chmielewski, L. (2015). A Semi-Parametric Approach for Side-Channel Attacks on Protected RSA Implementations. In *CARDIS*.

[Perin et al. 2014] Perin, G., Imbertl, L., Torres, L., Maurine, P., and Montpellier, R. A. (2014). Attacking Randomized Exponentiations Using Unsupervised Learning. In *CARDIS*.

[Preneel et al. 1993] Preneel, B., Govaerts, R., and Vandewalle, J. (1993). Hash functions based on block ciphers: A synthetic approach. In Stinson, D. R., editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 368–378. Springer.

[Riscure 2016] Riscure (2016). Riscure B.V. - Inspector SCA. `https://www.riscure.com/security-tools/inspector-sca`.

[Rivain 2011] Rivain, M. (2011). Fast and regular algorithms for scalar multiplication over elliptic curves. *IACR Cryptology ePrint Archive*, 2011:338.

[Rodrigues et al. 2016] Rodrigues, B., Pereira, F. M. Q., and Aranha, D. F. (2016). Sparse representation of implicit flows with applications to side-channel detection. In Zaks, A. and Hermenegildo, M. V., editors, *Proceedings of the 25th International Conference on Compiler Construction, CC 2016, Barcelona, Spain, March 12-18, 2016*, pages 110–120. ACM.

[Saeedi and Kong 2014] Saeedi, E. and Kong, Y. (2014). Fuzzy analysis of side channel information. *2014, 8th International Conference on Signal Processing and Communication Systems, ICSPCS 2014 - Proceedings*, pages 1–5.

[Schneider and Moradi 2016] Schneider, T. and Moradi, A. (2016). Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99.

[Sedra and Smith 1997] Sedra, A. S. and Smith, K. C. (1997). *Microelectronic circuits*, chapter 4. Oxford University Press, Inc., 4th edition.

[Shannon 1949] Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell Systems Technology Journal*, 28:657–715.

[Silberschatz et al. 2004] Silberschatz, A., Galvin, P. B., and Gagne, G. (2004). *Operating System Concepts*. Wiley.

[Trichina and Bellezza 2003] Trichina, E. and Bellezza, A. (2003). *Implementation of Elliptic Curve Cryptography with Built-In Counter Measures against Side Channel Attacks*, pages 98–113. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Tromer et al. 2010] Tromer, E., Osvik, D. A., and Shamir, A. (2010). Efficient Cache Attacks on AES, and Countermeasures. *Journal of Cryptology*, 23(1):37–71.

[Tunstall and Goodwill 2016] Tunstall, M. and Goodwill, G. (2016). Applying TVLA to Public Key Cryptographic Algorithms. Technical report, Eprint.

[Vaudenay 2002] Vaudenay, S. (2002). Security flaws induced by CBC padding - applications to ssl, ipsec, WTLS ... In Knudsen, L. R., editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 534–546. Springer.

[Vredendaal 2014] Vredendaal, C. (2014). Implementation of e-enumeration algorithm for DLP-based cryptosystems. http://scarecryptow.org/publications/sckangaroos.html.

[Walter 2001] Walter, C. D. (2001). Sliding Windows Succumbs to Big Mac Attack. In *CHES*, pages 286–299.

[Witteman et al. 2011a] Witteman, M., Jaffe, J., and Rohatgi, P. (2011a). Efficient side channel testing for public key algorithms: RSA case study. Technical report, CRI.

[Witteman et al. 2011b] Witteman, M. F., van Woudenberg, J. G. J., and Menarini, F. (2011b). Defeating {RSA} Multiply-Always and Message Blinding Countermeasures. pages 77–88.

[Witten and Frank 2011] Witten, I. H. and Frank, E. (2011). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[Woudenberg et al. 2011] Woudenberg, J. G. J. V., Witteman, M. F., and Bakker, B. (2011). Improving Differential Power Analysis by Elastic Alignment. In *CT-RSA*, pages 104–119.

[Yarom and Falkner 2014] Yarom, Y. and Falkner, K. (2014). FLUSH+RELOAD: A high resolution, low noise, L3 cache side-channel attack. In Fu, K. and Jung, J., editors, *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, pages 719–732. USENIX Association.

[Özgen et al. 2016] Özgen, E., Papachristodoulou, L., and Batina, L. (2016). Template attacks using classification algorithms. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 242–247.

Capítulo 4

Desafios de Segurança e Confiabilidade na Comunicação para Smart Grids

Yona Lopes, Tiago Bornia, Vitor Farias, Natalia C. Fernandes, Débora C. Muchaluat-Saade

Laboratório MídiaCom – Universidade Federal Fluminense (UFF) – Niterói, RJ – Brasil

Abstract

Smart grids will deeply change the way energy is delivered, from generation to consumers. In this new model, the amount of devices controlled and monitored dramatically rises allowing a more automated, smart, and efficient system. To ensure these goals, a network of highly secure communications, reliable, and that provides low delays is necessary so the monitoring and control of the grid can be performed correctly. However, the same interconnected system that transforms the old grid in a smart grid also brings new challenges to this scenario, such as security and reliability. Thus, the central focus of this chapter is the discussion of these challenges. We will discuss the key concepts related to smart grid, focusing on vulnerabilities and attacks that such network may suffer. Solutions and recommendations on key security challenges will also be addressed.

Resumo

A rede elétrica inteligente traz propostas que mudam de forma profunda a maneira como a energia é provida desde a geração até os consumidores finais. No novo modelo, a quantidade de dispositivos controlados e monitorados aumenta demasiadamente compreendendo inclusive o consumidor final, permitindo um sistema mais automatizado, inteligente e eficaz. Para tanto, será necessária uma rede de comunicação altamente segura, confiável e com baixos retardos, de forma que o monitoramento e o controle da rede elétrica possam ser realizados. No entanto, o mesmo sistema interconectado que torna a rede elétrica mais inteligente também traz novos desafios para este cenário, como segurança e confiabilidade. Assim, o foco central deste capítulo é a discussão sobre esses desafios. Serão abordados os principais conceitos relacionados a smart grid, com foco nas vulnerabilidades e ataques que esse tipo de rede pode sofrer. As soluções e recomendações relativas aos principais desafios de segurança também serão abordadas.

4.1. Introdução

Uma rede elétrica inteligente, conhecida como *Smart Grid*, traz propostas inovadoras que mudam de forma profunda a maneira como a energia é provida desde a geração até os consumidores finais [Lopes et al. 2015a]. O sistema elétrico tradicional, de forma geral, contava com uma comunicação que compreendia apenas parte do sistema, como as subestações e seus centros de controle, além da comunicação entre as subestações e entre os centros de controle. No novo modelo, a quantidade de dispositivos controlados e monitorados aumenta demasiadamente compreendendo inclusive o consumidor final, permitindo um sistema mais automatizado, inteligente e eficaz. A comunicação precisará dar o suporte para a estabilização das demandas e para a tarifação, garantindo uma resposta a demanda adequada e o livre mercado para compra e venda de energia em tempo real por consumidores finais. Para tanto, será necessária uma rede de comunicação altamente segura, confiável e com baixos retardos, de forma que o monitoramento e o controle da rede elétrica possam ser realizados.

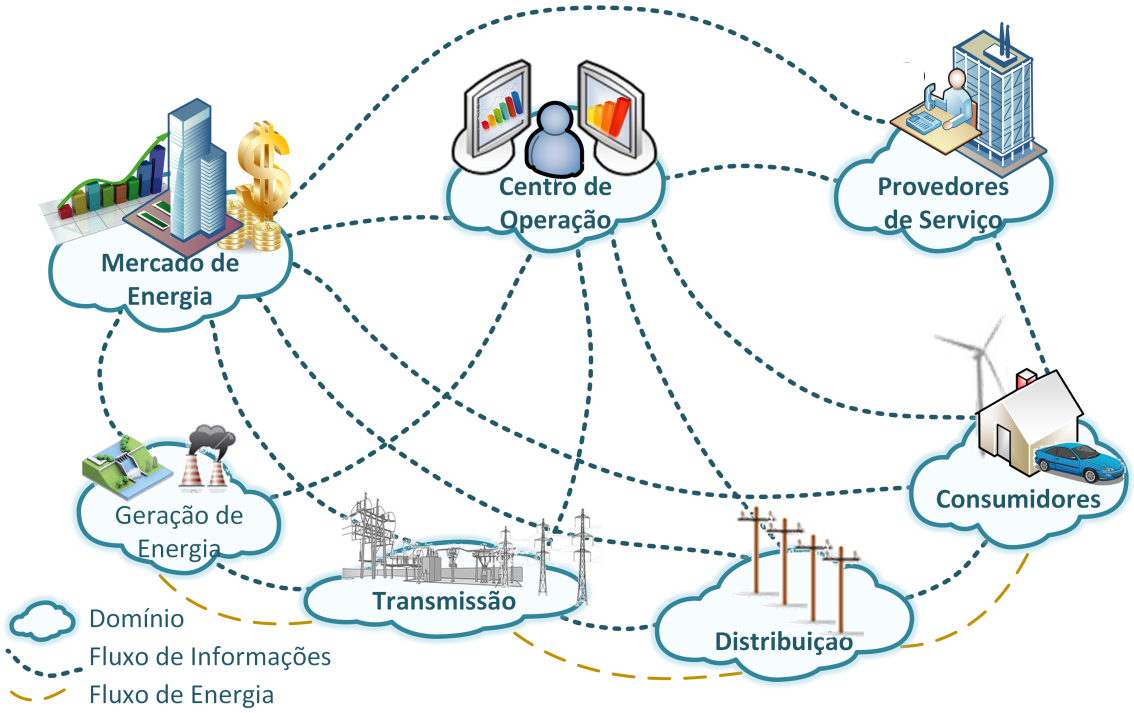


Figura 4.1. Atores das redes elétricas inteligentes e a comunicação entre eles, segundo o modelo proposto pelo NIST [NIST 2010, Lopes et al. 2015a].

De acordo com o modelo conceitual do NIST (*National Institute of Standards and Technology*) [NIST 2010], ilustrado na Figura 4.1, a rede elétrica inteligente é composta de sete domínios lógicos, com agentes e dispositivos inteligentes que devem ser interligados. Nesse novo cenário, os dispositivos finais da rede se tornam mais inteligentes e podem se comunicar diretamente com os centros de controle de dados. De fato, a implantação da rede elétrica inteligente começa com uma inserção em massa de medidores inteligentes. Além disso, o número de IEDs (*Intelligent Electronic Devices* – Dispositivos Eletrônicos Inteligentes) aumenta a fim de apoiar a Automação da Distribuição (*Distribution Automation* - DA). Em geral, a quantidade de dispositivos de automação, tais como

medidores inteligentes e IEDs, e a quantidade de dados coletados a partir desses dispositivos, aumentam significativamente. Dessa forma, a rede elétrica inteligente traz um enorme crescimento no volume de dados que deve ser gerenciado.

A fim de permitir uma implantação de sucesso das redes elétricas inteligentes, uma comunicação de rede robusta entre os dispositivos é necessária. Tal cenário envolve vários nós, enlaces, sistemas, protocolos e tecnologias para compor diferentes tipos de redes, formando uma arquitetura complexa e ampla. Este sistema interligado traz várias vantagens, tais como a visibilidade, a disponibilidade e o controle remoto que tornam possíveis várias novas operações para a concessionária, tornando o sistema mais inteligente. Além disso, novas aplicações de energia, tais como o planejamento de capacidade e o controle do horário de pico, irão melhorar o sistema. Também, novas aplicações facilitarão a implantação de novos serviços de energia, tais como sistemas para auditorias do uso de energia, programas de resposta à demanda e pontos de carregamento de veículos elétricos [Budka et al. 2014].

No entanto, o mesmo sistema interligado que permite a criação de diversas novas aplicações no provimento de energia elétrica também traz ameaças à segurança e faz com que todo o sistema fique vulnerável a ataques. No passado, as redes de comunicação para instalações elétricas ficavam restritas a áreas fechadas e seguras, como em subestações, que garantiam a segurança física da rede. Devido à integração com medidores inteligentes, nuvens, e outras fontes de informação, a segurança física para o acesso à rede não é mais uma opção, o que pode comprometer o controle do sistema elétrico.

Portanto, as redes elétricas inteligentes não podem avançar sem lidar com os problemas relacionados à segurança. Os ataques contra a rede de energia elétrica podem impactar diretamente a população e afetar as pessoas, o comércio, as empresas e qualquer um que não possa ficar sem energia elétrica. Qualquer possibilidade de evento que cause impacto na confidencialidade, na integridade e na disponibilidade dos domínios da rede elétrica inteligente é considerada uma ameaça.

Ataques que tentam ganhar vantagem em cima das vulnerabilidades encontradas em sistemas que trocam informações em uma rede são conhecidos como *data-centric threats* [Wei and Wang 2014]. Essas ameaças podem ser difíceis de detectar e podem resultar em danos críticos para a infraestrutura industrial. Um *worm* pode reprogramar uma instalação de controle industrial para degradar o equipamento e gerar logs de operação falsos, comprometendo a manutenção. Um atacante pode assumir o controle do sistema ou roubar informações confidenciais mesmo sem acesso físico à planta [Wei and Wang 2016]. Ataques contra instalações nucleares, como o primeiro worm descoberto em sistemas industriais como o Stuxnet [Falliere et al. 2011] e o ataque ao sistema elétrico ucraniano [Assante 2016], são uma demonstração do potencial destrutivo de ameaças cibernéticas.

Por exemplo, o SCADA (*Supervisory Control and Data Acquisition*), que é um sistema muito importante usado para supervisionar e controlar a operação do sistema elétrico, deve ser interligado com toda a estrutura de rede. Vulnerabilidades do sistema SCADA são geralmente correlacionadas ao uso de uma Interface Homem-Máquina (IHM) e os históricos de dados [Wilhoit 2013]. Históricos de dados são bancos de dados de *logs* que armazenam as tendências e informações históricas sobre os processos de um sistema de controle industrial. Se o atacante puder comprometer a IHM, ele terá acesso a áreas

seguras onde ele pode modificar o ajuste de dispositivos ou controlar equipamentos. Uma abertura ou fechamento indevido de um disjuntor pode causar uma interrupção no fornecimento de energia de forma desnecessária. Além disso, caso um circuito esteja em manutenção, um fechamento indevido de um disjuntor poderia ameaçar a vida humana. Da mesma forma, se um invasor puder acessar o histórico de dados, poderá ler o banco de dados centralizado com todas as informações de registro sobre o ambiente do sistema de controle industrial. Assim, o atacante terá informações sobre os sistemas de segurança, bem como uma lista de comandos usados em dispositivos como IEDs e Controladores Lógicos Programáveis (*Programmable Logic Controller* - PLC) e IEDs. Não só o SCADA é vulnerável a ataques, mas também medidores inteligentes e quaisquer IEDs. É importante notar que os medidores inteligentes estão nas residências dos clientes, que são potenciais atacantes. Estes poderiam alterar os dados de consumo, divulgar informações relacionadas à privacidade e usar medidores inteligentes como um ponto de entrada para grandes ataques. Portanto, interrupções e danos causados por ataques passivos ou ativos tornam-se uma ameaça real. As motivações para os ataques variam desde a redução de custos na conta de energia à promoção do terrorismo.

Este capítulo aborda as questões de segurança relacionadas às redes elétricas inteligentes, suas principais vulnerabilidades e ameaças. Assim, descrevem-se as principais falhas de arquitetura que tornam o sistema elétrico vulnerável a ataques para causar interrupções de energia, furto de energia e quebra de privacidade.

O restante do texto está organizado da seguinte forma. Inicialmente, uma breve descrição das redes elétricas inteligentes é feita na Seção 4.2. Em seguida, na Seção 4.3, os principais conceitos de segurança relacionados com as redes elétricas inteligentes são detalhados. Os ataques mais comuns relacionados a *smart grids*, considerando tanto ataques às subestações quanto ataques à Infrastrutura de Medição Avançada (AMI), são detalhados na Seção 4.3.2. As principais soluções atuais para criar um ambiente seguro para as comunicações de *smart grids* são apresentadas na Seção 4.4. Por fim, as últimas seções do capítulo apresentam as conclusões e direções futuras para pesquisas na área.

4.2. Redes Elétricas Inteligentes

Devido ao crescente aumento populacional e ao aumento do número de equipamentos em uso nas residências, a demanda por energia tem crescido cada vez mais nos últimos anos [Lopes et al. 2015a]. No entanto, para acompanhar esse crescimento, o setor precisa investir muito em infraestrutura. Um caminho, usado por muito tempo, foi o investimento no aumento da infraestrutura para geração de energia, com a construção de novas usinas geradoras para suprir essa demanda. Contudo, a regulamentação para as construções e demandas ambientais muitas vezes atrasam e/ou impedem esse tipo de construção. Essas caraterísticas resultam na necessidade de estudo e implementação de novos mecanismos e sistemas para suprir o aumento da demanda sem a construção de novas usinas geradoras. Assim, a modernização da infraestrutura existente e o desenvolvimento de novas propostas ganharam força nos últimos anos. Outro ponto importante é que o sistema elétrico já vinha passando pela necessidade de modernização, já que pouca inovação tinha sido feita nas últimas décadas. Consequentemente, os equipamentos utilizados e as tecnologias, algumas vezes, eram os mesmos de 40 anos atrás [Gungor et al. 2011]. Os medidores residenciais tradicionais, por exemplo, precisam de funcionários para realizar a leitura do

consumo e o corte/religamento de energia, o que gera um custo alto para a empresa concessionária. Desta necessidade inevitável de modernização do sistema elétrico, surgiram as redes elétricas inteligentes, ou *Smart Grids*, tornando imprescindível a implantação de um sistema de comunicação mais "inteligente" [Lopes et al. 2012].

Esta modernização vem causando uma grande revolução nas redes de energia elétrica, aumentando os ganhos em confiabilidade, eficiência energética, participação dos consumidores e geração de uma energia mais limpa [Patel et al. 2011]. Tal revolução está ocorrendo porque as redes elétricas inteligentes são baseadas em conceitos como a monitoração inteligente de todos os dispositivos do sistema e a transmissão dos fluxos de comunicação e de energia de forma bidirecional, cenário bastante distinto do tradicional. Dentre as novas propostas, destacam-se a geração de energia de forma distribuída, o amplo uso de fontes renováveis, o uso de carros elétricos, um intenso monitoramento da rede elétrica, o uso de medidores inteligentes, entre outros. Com as redes elétricas inteligentes, o consumidor passa a ser parte fundamental do funcionamento e controle da rede elétrica. Os consumidores, que no sistema tradicional apenas consomem energia, podem ter nesse novo modelo também o papel de produtor de energia elétrica. Além disso, os medidores inteligentes localizados nas residências passam a gerar uma quantidade enorme de informação que poderá ser usada para o gerenciamento e controle do sistema. Portanto, para que o desenvolvimento da rede elétrica inteligente seja possível, a inteligência e as tecnologias como as de supervisão, controle e proteção, antes existentes apenas em parte do sistema elétrico, se tornam imprescindíveis da geração até o consumidor final [Lopes et al. 2015a].

Um ponto importante é que esse novo sistema elétrico depende de uma sofisticada infraestrutura de redes de comunicação para dar suporte à comunicação entre os dispositivos inteligentes que monitoram e atuam na rede. Além disso, é necessário dar suporte às empresas de distribuição de energia e aos usuários, que podem consumir ou gerar energia [Budka et al. 2010]. Novas necessidades surgem, como o aumento da confiabilidade da rede, o aumento da eficiência operacional da rede, a melhora da qualidade para o consumidor e o aumento da variedade dos serviços providos. Porém, todas essas melhoras trazem diversos desafios para as redes de comunicação e um dos mais importantes, se não o mais importante deles, é o provimento de segurança. Conhecer as principais áreas chaves da rede elétrica inteligente ajuda a entender os problemas e desafios na área de segurança e suas possibilidades de solução.

4.2.1. Domínios das Redes Elétricas Inteligentes

O modelo conceitual das redes elétricas inteligentes foi proposto pelo NIST [NIST 2010] e é ilustrado na Figura 4.1 com fluxos bidirecionais de informação. O NIST divide o modelo em sete domínios que, juntos, representam a comunidade de redes inteligentes de interesse. Esses domínios são:

- Domínio de geração de energia: composto pelas tradicionais plantas de geração e pelo armazenamento de energia. Para que possa trocar informações sobre a energia gerada ou armazenada, o domínio de geração troca dados com o domínio da operação da rede elétrica e com o domínio do mercado de energia.

- Domínio dos consumidores: além da funcionalidade de consumo de energia, a geração de energia em pequena escala e o armazenamento de energia também se encontram nesse domínio. Para isso, o domínio dos consumidores se comunica com os domínios de operação da rede e de mercado de energia.

- Domínio de Distribuição e Domínio de Transmissão: os sistemas de transmissão e distribuição passam a ser muito mais ativos, trocando informação com a operação da rede elétrica, com consumidores e seus medidores inteligentes e com o mercado de energia.

- Domínio de provedores de serviços: se comunica com os consumidores para faturamento, operações de resposta à demanda e serviços de terceiros. Para obter informações de medições e controle da rede elétrica, se comunica também com o domínio de mercado e de operação da rede elétrica.

- Domínio do mercado de energia (atacado, varejo e comércio): é responsável pelo balanceamento de oferta e demanda de energia e, portanto, coleta e envia informações de oferta e demanda aos domínios de geração, provedores de serviços e operação da rede elétrica inteligente.

- Domínio da operação da rede elétrica: se comunica com todos os outros domínios a fim de coletar os dados para garantir o controle e a operação eficiente do sistema.

Esses domínios foram ilustrados na Figura 4.1 e se comunicam entre si conforme mostrado na figura.

Apesar de serem domínios separados, são intimamente relacionados. Uma aplicação de um domínio pode interferir na outra, pode necessitar de dados de outros domínios, pode se comunicar com outra aplicação, etc. Além disso, aplicações tradicionais deverão coexistir com as novas aplicações advindas das redes elétricas inteligentes (as áreas que darão origem a essas aplicações são descritas na Seção 4.2.2) e/ou evoluir para acompanharem as mudanças e novas aplicacões. Exemplos de aplicações tradicionais são a teleproteção e o SCADA também chamado de *software* supervisório. A teleproteção usa um sistema de comunicação entre duas subestações. Com isso, se um equipamento de proteção em uma subestação detecta uma falha em uma extremidade, a outra extremidade é notificada e ações de proteção são iniciadas a fim de isolar a falha. Já o sistema SCADA, que no passado era suportado por mainframes e sistemas fechados de fornecedores, atualmente, faz uso da rede de comunicação para interconectar todos os equipamentos das subestações que são supervisionados por ele. O SCADA é utilizado para supervisionar, controlar, otimizar e gerenciar os sistemas de geração e transmissão de energia elétrica. Já o SCADA de nova geração deve ser adaptado para um cenário com maior granularidade de supervisão e novas possibilidades. Dentre os benefícios trazidos pelos sistemas SCADA de nova geração, destacam-se a análise de consumo e demanda, a análise da carga dos consumidores, a verificação de falhas, o rearranjo da topologia, a análise da carga nos transformadores, a medição inteligente, entre outros [Lopes et al. 2012]. Com a evolução para as redes elétricas inteligentes, o SCADA incorporará novos elementos inteligentes, tais como: unidades de medição fasorial, relés inteligentes, novas fontes de

geração de energia com utilização de fontes renováveis, armazenamento de energia em veículos elétricos (EV), medidores inteligentes, etc [Giani et al. 2011].

Ao permitir geração de energia pelo consumidor, uma rede elétrica inteligente promove uma estreita relação entre compradores e vendedores, clientes e concessionárias. Um fluxo bidirecional de energia e comunicação bem como as capacidades *plug-and-play* são seu objetivo final e permitirão que várias tecnologias possam fornecer, entregar e utilizar os recursos de forma confiável, eficiente e segura.

4.2.2. Áreas chaves das Redes elétricas inteligentes

Como abordado anteriormente, as aplicações tradicionais, como a teleproteção e o SCADA, deverão coexistir com as novas aplicações advindas das redes elétricas inteligentes. Estas aplicações surgem de áreas chaves que permitem o desenvolvimento de novos sistemas. Dentre as áreas existentes estão a infraestrutura de medição avançada (AMI - *Advanced Metering Infrastructure*), a microgrid, a planta de energia virtual (VPP- *Virtual Power Plant*), o gerenciamento pelo lado da demanda (DSM - *Demand Side Management*) e a resposta à demanda (DR - *Demand Response*), detalhados a seguir [Lopes et al. 2012].

4.2.2.1. Infraestrutura de Medição Avançada

A AMI (*Advanced Metering Infrastructure*) é um sistema integrado composto por medidores inteligentes, infraestrutura de comunicação e sistemas de gerenciamento capazes de permitir comunicação bidirecional entre medidores e concessionária. A AMI visa permitir diversas facilidades para consumidores residenciais, comerciais e industriais. Sua infraestrutura será detalhada na Seção 4.3.3.1.

As concessionárias geralmente iniciam a implantação das redes elétricas inteligentes pela AMI. Isso se deve, principalmente, a necessidade de informações, monitoramento e comunicação bidirecional entre consumidores e concessionária. A AMI vai além da medição de energia periódica, gerando dados que são usados por outras aplicações ou domínios das redes elétricas inteligentes. Por exemplo, as medidas fornecidas pelos medidores inteligentes também são usados para dar suporte às aplicações de tarifas em tempo real (*Real Time Pricing* - RTP), tarifas horo-sazonais (*Time of Use* - TOU) e tarifa de picos críticos (*Critical Peak Pricing* - CPP), ferramentas usadas para tarifação e para resposta a demanda [Budka et al. 2010] [1]. Mecanismos de tarifação dinâmica, como o TOU e o CPP, contribuem para uma implementação da resposta a demanda eficiente, o que contribui para uma possível redução de custos. Além disso, o medidor pode fazer parte de um domínio de geração quando na casa do consumidor tiver uma geração local. Assim, as informações oriundas do gerador residencial, parte de uma rede de geração distribuída (GD), poderão ser trocadas através do uso do medidor inteligente e da AMI.

Com a AMI, torna-se possível a detecção de falhas na rede elétrica de distribuição e a detecção de furtos de energia. Além disso, quando a AMI está trafegando informações de alguma GD, torna-se parte do sistema de proteção e controle, tornando possível, por

[1] Resposta a Demanda, do inglês *Demand Response*, é a iniciativa de alterar temporariamente o consumo de energia em resposta às condições de fornecimento de energia ou aos eventos na rede [EPRI 2009].

exemplo, o isolamento de falhas nos sistemas elétricos. Outra vantagem é que a AMI permite que eletrodomésticos respondam a sinais de preço, aumentando ou diminuindo o consumo de acordo com as variações de mercado. Com a AMI, o religamento e o corte de energia podem ser feitos de forma remota, além de permitir o acompanhamento do consumo. Em resumo, a AMI permite às concessionárias de serviços públicos:

- obter uma leitura automática e remota (telemetria) e faturamento precisos;
- controlar remotamente o corte e religamento do fornecimento de energia;
- detectar interrupções na distribuição de energia;
- tarifar em tempo real e fazer a tarifação horo-sazonal (TOU);
- fornecer outras medições como água e gás;
- impedir a manipulação indevida de leituras e dados de faturamento;
- melhorar o serviço de suporte ao consumidor final através de comunicação em tempo real; e
- possibilitar a implementação do sistema de proteção e controle dos dispositivos sem a necessidade de implementação de nova infraestrutura.

A implementação da AMI, com informações em tempo real, contribui para maior retorno de investimento e custo operacional mais baixo, o que justifica o investimento em longo prazo. Outra aplicação de grande prioridade é a resposta à demanda, que usa os dados gerados em tempo real para tomar ações com relação à geração de energia. Nesse sentido, ter uma rede confiável, com informações disponíveis em tempo real, torna-se uma premissa básica para entrega de energia confiável para os usuários finais. A grande maioria das falhas no provimento de energia podem ser evitados ou contornados pelo monitoramento em tempo real, pelo diagnóstico e proteção da rede, que precisam ser confiáveis e seguros.

4.2.2.2. Microgrids

A *microgrid* é um novo paradigma que consiste na criação de pequenos sistemas elétricos localizados e compostos por geração, armazenamento e cargas com a ideia de ser autossuficiente. É um novo paradigma que pode combinar vários Recursos Energéticos Distribuídos (DER - *Distributed Energy Resources*) para formar um todo. As unidades de DER são as fontes geradoras de energia que podem ser compostas por unidades de geração distribuída e por unidades de armazenamento distribuído, incluindo veículos elétricos [Lopes et al. 2015a].

Assim, esse conceito inclui GD, armazenamento de energia, conexão entre GD e rede externa de energia, e mecanismos de controle [Pan et al. 2014]. Várias *microgrids* interligadas, de acordo com o conceito *plug and play*, podem criar uma rede macro, chamada de *macrogrid* [Lopes et al. 2012]. Com isso, o controle de uma *microgrid* deve considerar três camadas, sendo elas o fluxo de informação, o fluxo de tensão e a camada física (real), mostrada na Figura 4.2. Embora a *microgrid* opere principalmente ligada à rede de distribuição [Bayod-Rújula 2009], ela pode operar na forma de "ilha", onde a própria energia gerada pela geração distribuída supre a necessidade da demanda [Lopes et al. 2012]. Esse modo, também chamado de ilhamento, faz com que a *microgrid* funcione de forma autônoma, desligada da rede externa. Esse modo proporciona continuidade do fornecimento em caso de falhas na rede externa. Nesse caso, a *microgrid* pode ser ressincronizada com o macrossistema após a restauração da rede externa [Bayod-Rújula 2009]. Dessa forma, as *microgrids* podem melhorar a confiabilidade no fornecimento de energia, pois se baseiam na premissa de que a geração de energia, ou a maior parte dela, está próxima ao consumidor e restrita a uma área menor.

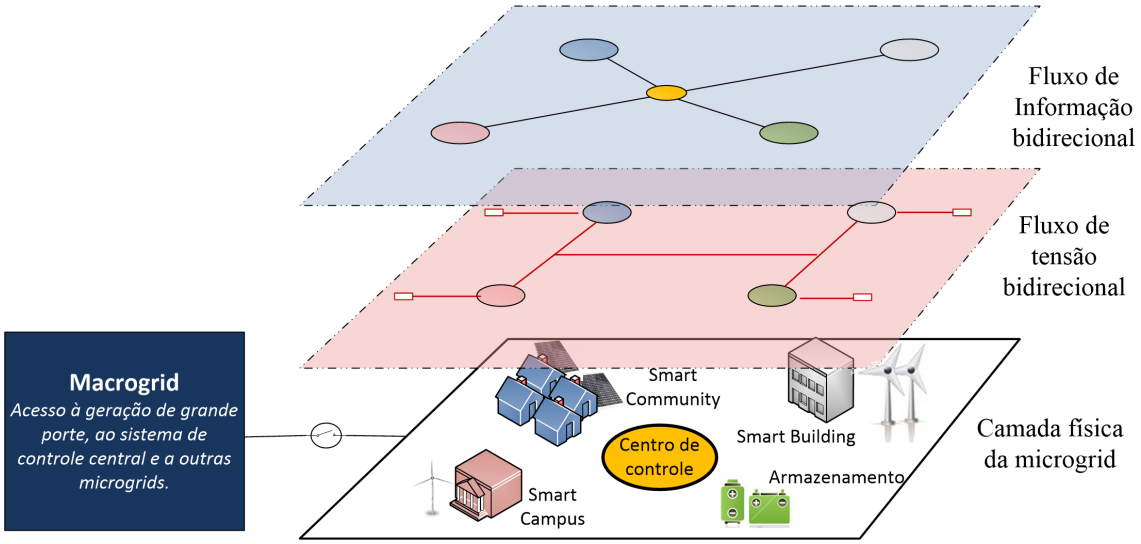


Figura 4.2. Exemplo de uma *microgrid*, onde a comunição e a distribuição elétrica coexistem, interligando as diversas fontes de geração distribuídas [Lopes et al. 2012].

Dentro do contexto de uma *microgrid*, as fontes de energia podem ser de geradores ou ainda de bancos de armazenamento de energia. Um tipo de banco de armazenamento que pode ser de grande utilidade no momento de uma falha do fornecimento são as baterias dos carros elétricos.

É desafiadora a necessidade de tornar o lado do consumidor mais inteligente, mais eficiente e rentável. Especialmente no futuro, a GD e as *microgrids* serão muito comuns com casas e prédios fazendo uso da energia renovável. Quando a capacidade das fontes geradoras exceder a própria demanda, o restante de energia deverá ser exportada para a *microgrid* e a *macrogrid*. Uma programação dinâmica e otimizada destes geradores distribuídos pode alimentar as demandas e reduzir o custo total, além de alcançar uma maior eficiência energética em escala.

Em uma *microgrid*, são usados controladores, que são dispositivos que são co-

nectados aos geradores e às cargas para controlar o funcionamento destes. As cargas são quaisquer dispositivos elétricos conectados à rede que necessitem de energia elétrica para funcionar, ou seja, os consumidores de energia. As cargas podem ter características bem diferentes, podendo ser usuários residenciais, comerciais ou industriais.

Ressalta-se que a *microgrid* tem seus próprios requisitos de controle entre geradores e consumidores de energia devido à sua escala limitada. Os métodos de controle utilizados dentro das *microgrids* podem ser centralizados, distribuídos ou hierárquicos ou métodos que combinam vários tipos. O mecanismo de controle deve permitir a adição e remoção flexível de geradores distribuídos em um estilo *"plug-and-play"*, sem perturbar o resto do sistema ou sem a necessidade de reconfigurar todo o sistema. O controle também pode atribuir diferentes prioridades às cargas, que podem ser priorizadas de acordo com a sua importância como mais ou menos críticas.

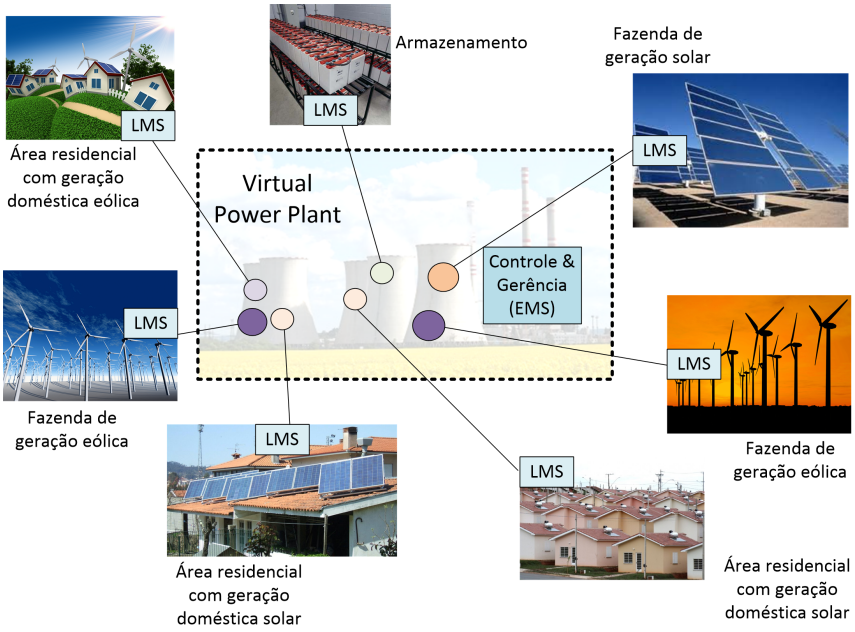
4.2.2.3. VPP (*Virtual Power Plant*)

Uma VPP, também conhecida como *Virtual Utility*, pode ser definida como um novo modelo de infraestrutura de energia que consiste na integração de diferentes tipos de GD controlados por um sistema de gerenciamento de energia (*Energy Management System* - EMS). A rede é composta por um controle centralizado de diferentes grupos de geração distribuída, chamados de *clusters*. Cada um destes *clusters* é controlado por uma estação de gerenciamento local (*Local Management Station* - LMS) e cada LMS tem informações sobre os requisitos de energia dos usuários conectados ao seu *cluster*, como eletricidade, nível de água no tanque, etc [Bayod-Rújula 2009]. Esse sistema é ilustrado na Figura 4.3.

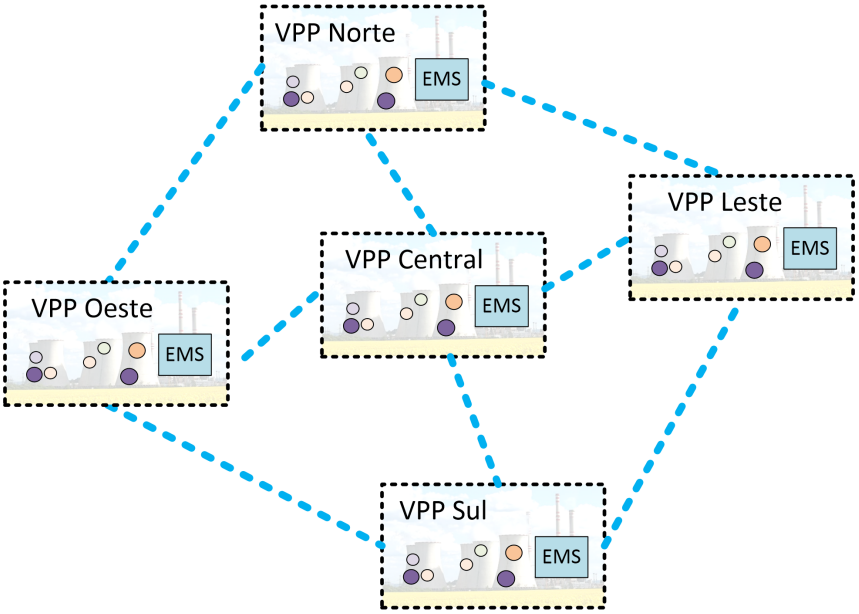
O EMS recebe as informações de cada LMS e define a entrada ou saída de energia de cada *cluster* na rede. Com a informação da EMS, o LMS configura o cluster para que ele entre em funcionamento ou fique em *standby*. Além disso, o EMS pode priorizar o uso de recursos de energia distribuídos (DER) ao invés do uso de combustíveis fósseis.

Os benefícios da VPP estão relacionados à otimização do rendimento de utilização de toda a rede, à alta confiabilidade da produção de energia, ao controle total da rede para atingir o principal objetivo da EMS, à alta velocidade necessária para acompanhar as mudanças rápidas na demanda do sistema e à alta integração dos DER [Bayod-Rújula 2009].

Para um operador de rede ou concessionária de energia, a compra de energia a partir de uma VPP é equivalente a compra a partir de uma planta convencional. O conceito de VPP não é por si só uma nova tecnologia, mas sim um método de organização de geração e armazenamento descentralizado de uma forma que maximiza o valor da energia gerada para a concessionária. A VPP usando GD, DER e armazenamento de energia tem potencial para substituir a planta convencional [Bayod-Rújula 2009].



(a) Modelo de VPP, com o controle local e central das várias fontes geradoras.



(b) Agregação de GD com resposta à demanda, através da gestão de diversas VPPs integradas.

Figura 4.3. Conceito de *Virtual Power Plant* (VPP) [Lopes et al. 2015a].

4.2.2.4. Gerenciamento pelo lado da demanda (DSM) e a resposta à demanda (DR)

Segundo o EPRI (*Electric Power Research Institute*), resposta à demanda (*Demand Response* - DR) é uma mudança temporária no consumo de energia em resposta às condições de fornecimento de energia ou aos eventos na rede [EPRI 2009]. A inclusão de novas fontes de energia e elementos de armazenamento combinados com a necessidade de re-

duzir os picos de carga impulsionaram a introdução de aplicações de resposta à demanda. Para isso, incentivos monetários podem ser usados de modo a evitar preços elevados de energia. Essas aplicações objetivam prover confiabilidade através de uma série de ações que visam reduzir a carga da rede no horário de pico, quando a concessionária está perto da sua capacidade máxima. Por exemplo, pode-se reduzir a quantidade de energia consumida pelos aparelhos durante o período de pico de potência, evitando inclusive apagões. Nesse cenário, o cliente passa a ter um papel ativo no fornecimento de energia elétrica. Esse sistema permite que consumidores transfiram o consumo de energia para momentos fora do horário de pico, tomando vantagem do preço da energia em tempo real, das informações da rede, controle da carga, etc [Bayod-Rújula 2009].

Conceitualmente, a resposta à demanda é equivalente ao aumento de geração no processo de equilíbrio do sistema. A solução de reduzir o uso de energia e utilizar a geração distribuída quando a oferta de energia é baixa tem ganhado cada vez mais aceitação no mercado. A DR e a DSM reduzem a carga e acrescentam a capacidade de geração em caso de emergência.

A DR, muitas vezes, usa a GD de forma que a energia passe a ser provida de um ponto mais próximo do consumo ou passe a receber energia de outras fontes conectadas à rede. Assim, em alguns casos, a DR pode não só reduzir o consumo global de energia, mas também mudar a origem da geração para uma GD. Ressalta-se que para que seja possível a implementação da DR, outro *driver* da rede elétrica inteligente precisa ser implementado: a automação da distribuição (DA). A DA é a ideia de se estender o monitoramento e controle da rede até a distribuição, de forma que dispositivos que antes não eram automatizados passem a ser. Atualmente, empresas de energia estão acostumadas com a gestão de um número limitado de pontos de monitoramento e controle, por exemplo, centenas de subestações. Novas tecnologias de comunicação devem ser introduzidas na distribuição a fim de conectar dezenas de milhares de *endpoints* encontrados na automação da distribuição.

4.2.2.5. As Áreas Chaves e o Desafio de Provimento de Segurança

A implementação das novas áreas advindas das redes elétricas inteligentes resulta no aumento do número de usuários com diferentes níveis de confiabilidade cooperando entre si e atuando no sistema, tornando o provimento de segurança uma questão crucial [Neuman and Tan 2011, Zhu et al. 2011]. Entre os mecanismos necessários para o provimento de segurança, destacam-se a autenticação das solicitações dos usuários, a autenticação de mensagens enviadas por aparelhos inteligentes, como mensagens de oferta de energia de fontes alternativas [Yan et al. 2011], e as métricas para avaliar a importância das informações trocadas entre usuários e fornecedores na rede [Chim et al. 2011]. A troca de mensagens tem impacto em todo o controle e gerência da rede, de forma que a autenticidade e a confiabilidade dos dados trocados devem ser sempre asseguradas pela infraestrutura da rede elétrica inteligente. A segurança na comunicação também diz respeito à confidencialidade dos dados da rede e dos usuários, tais como informações de endereço e de cartão de crédito [Lopes et al. 2015a].

Especialmente no modelo de *microgrid*, todas as casas podem fornecer e consu-

mir energia da *microgrid*, de tal forma que os fluxos energéticos podem fluir bidirecionalmente e ser dinamicamente reconfigurados. É importante que o sistema da *microgrid* funcione de forma distribuída, mas evitando que mensagens falsas sejam inseridas na rede com o fim de prejudicar a distribuição de energia ou a cobrança posterior ou, ainda, que informações sejam roubadas para ferir a privacidade dos usuários. Em particular, os roteadores das *microgrids* podem utilizar enlaces sem fio, os quais são mais susceptíveis a ataques do que redes cabeadas [Lopes et al. 2012]. Portanto, a segurança das *microgrids* e de seus roteadores deve contar com mecanismos de controle de acesso, gerência de chaves e certificados, detecção de intrusão e de mau comportamento, entre diversos outros [Zhu et al. 2011].

Para que os dados da GD e da AMI sejam trocados, os medidores inteligentes precisam estar conectados à rede, enviando e recebendo mensagens. Uma vez que segurança é uma preocupação, é natural supor que os medidores serão equipados com as técnicas de segurança padrão, tais como utilização de certificados digitais e criptografia [Lopes et al. 2012]. Contudo, já é sabido que o uso dessas técnicas não é suficiente para impedir que o sistema seja atacado [Cleveland 2008]. Vide a Internet, a qual está munida dessas e outras técnicas, mas ainda sofre com frequentes problemas de segurança, em especial os ataques de negação de serviço [Wang et al. 2011]. O volume de ataques está fortemente correlacionado com a quantidade de *hackers* espalhados pelo mundo. Muito embora muitos *hackers* ajam maliciosamente para obter vantagens, muitos são adolescentes querendo quebrar novas barreiras. No contexto das redes elétricas inteligentes, a preocupação é relativa à qual impacto que esses *hackers* teriam sobre a rede elétrica, uma vez que tiverem dentro de suas casas medidores inteligentes capazes de interferir ativamente no funcionamento do sistema. Os incentivos para criar ataques na rede vão desde conseguir mudar contas de luz até conseguir causar apagões em cidades inteiras [Rahman et al. 2012]. Dessa forma, a segurança na AMI interfere não apenas no gerenciamento doméstico da energia, mas também na segurança dos controles de automação das subestações em *grids* e *microgrids* [Lopes et al. 2012].

Outro fator chave para segurança é que as demandas para automação de sistemas de energia controlados por computador têm crescido enormemente devido a sua importância [Cheung et al. 2007]. Os IEDs, que participam da proteção, do controle e da automação do sistema elétrico, têm uma comunicação autônoma na rede e podem evitar que uma falha no sistema elétrico de potência seja propagada causando, por exemplo, um apagão. No entanto, caso tenham um mau funcionamento, devido a um ataque por exemplo, também podem causar uma falha. O sistema tem que ser seguro o suficiente para que o acesso desses dispositivos por terceiros mal intencionados não seja permitido. Por exemplo, um pequeno atraso na transmissão de dados para operar o equipamento de proteção pode resultar em falha na subestação [Cheung et al. 2007].

Muito tem se pesquisado na área de segurança para *smart grids*, mas, devido a sua importância e ampla gama de possibilidades de ataque, é uma das áreas com mais desafios e oportunidade de pesquisa.

4.3. Segurança da Informação em *Smart Grids*

A compreensão do impacto dos ataques sobre as comunicações da rede elétrica depende da compreensão do conceito de incidente cibernético. De acordo com FIPS (*Federal Information Processing Standards*), este conceito é definido da seguinte forma:

> "Uma ocorrência que ponha em risco real ou potencial, a confidencialidade, integridade ou disponibilidade de um sistema de informação ou a informação dos processos que o sistema, armazena ou transmite ou que constitua uma violação ou ameaça iminente de violação de políticas de segurança, procedimentos de segurança, ou políticas de uso aceitável."[PUB 2006]

Recentemente, a discussão sobre as ameaças de segurança cibernética contra as redes de energia elétrica aumentou e tornou-se uma questão fundamental para *smart grids*. A integração de modelos de informação com as redes de comunicação para sistemas de energia trouxe novos desafios de segurança relacionados com a autenticidade, confidencialidade, integridade e disponibilidade. A interconexão de dispositivos que são distribuídos em locais sem segurança física é uma das principais preocupações. Por exemplo, o uso de uma AMI configura uma ameaça especial, pois os usuários finais são capazes de introduzir diretamente informação no sistema. A invasão de medidores por usuários, vírus ou *hackers* executando ataques de negação de serviço poderia interromper o fornecimento de energia para uma cidade inteira.

Os ataques contra *smart grids* podem ser devastadores, pois incluem toda a rede de energia, compreendendo desde as subestações e redes de distribuição até as residências e instalações comerciais e industriais. As consequências dos ataques variam de falhas no serviço até danos físicos, no caso em que um atacante é capaz de perturbar o sistema de proteção, comprometendo a segurança em instalações elétricas. As soluções para essas ameaças ainda estão em discussão e podem incluir técnicas de segurança bem conhecidas já aplicadas na Internet e novos protocolos de segurança para comunicações em *smart grids*. Para proporcionar uma melhor compreensão desses ataques e suas medidas preventivas, os principais conceitos de segurança relacionados a este cenário serão discutidos: autenticação, autorização, responsabilização, privacidade, integridade, disponibilidade e proteção física.

- Autenticação

  A autenticação é a capacidade de verificar a identidade de uma entidade. Em redes inteligentes, todas as entidades no sistema devem ter uma identificação verificável. Isto significa que todas as partes envolvidas, como usuários, empresas, IEDs, sensores, dispositivos domésticos, medidores inteligentes, carros elétricos, etc., devem ser identificados de forma exclusiva no sistema de um modo seguro. Na Internet, isto pode ser conseguido pela utilização de uma infraestrutura de chave pública (PKI - *Public Key Infrastructure*) ou sistemas de identificação mais simples no caso de dispositivos menos sensíveis. Esta tecnologia poderia ser aplicada de um modo simples para as redes inteligentes. Por exemplo, os sistemas de supervisão e controle, tais como o SCADA, que têm uma função importante para automação de subestações e podem controlar os dispositivos de campo, devem ser identificados pelo

uso de uma certificação digital para evitar ataques *man-in-the-middle*. Isso implica a compra de um certificado publicamente verificável. Este tipo de investimento, no entanto, não é justificável para dispositivos dentro de uma rede doméstica. Neste caso, onde não espera-se que os dispositivos interajam com sistemas sensíveis, mas apenas forneçam informações e serviços úteis para o usuário, os dispositivos podem ser autenticados por certificados autoassinados ou esquemas de login/senha. Um bom sistema de autenticação é a base principal para fornecer todos os outros conceitos de segurança. A simples utilização de uma comunicação cifrada não fornece a segurança necessária, porque os usuários mal intencionados podem falsificar identidades e danificar o sistema. Assim, a capacidade de verificação da identidade da outra entidade é um fator chave para a implementação de um ambiente seguro.

- Autorização

  O conceito de autorização está intimamente relacionado com o conceito de autenticação. Ele representa a capacidade de verificar as políticas do sistema para conceder ou não o acesso de uma entidade autenticada para um sistema específico. Sistemas de autorização diferem na granularidade de políticas. Uma política muito simples seria a de conceder acesso a todos os usuários autenticados. Por exemplo, em uma rede doméstica, todos os dispositivos que foram registrados pelo usuário devem ser capazes de se conectar ao sistema de gestão da casa. No entanto, esta política não se encaixa em um acesso ao sistema de subestação, onde há usuários com diferentes níveis de prioridade, como visualização apenas ou autorização para alterar parâmetros. Além disso, os dispositivos podem ter autorização para interagir apenas com um conjunto pré-determinado de dispositivos, e assim por diante. Nestes casos, políticas relacionadas a atributos, papéis, tempo, etc., são necessárias.

- Responsabilização

  Outra questão importante para garantir a segurança em um sistema de comunicação é a capacidade de registrar eventos. Assim, sempre que um evento incomum acontece, o administrador do sistema deve ser capaz de rastrear quais eventos anteriores conduziram à situação. Isso é especialmente importante para realizar a auditoria e descobrir as causas de ataques, quando eles acontecem. Essa é a essência dos sistemas de responsabilização. Uma característica importante é que os logs do sistema devem ser armazenados e protegidos, a fim de evitar que um usuário subverta a informação registrada para encobrir a ação maliciosa.

- Privacidade

  Privacidade é outro requisito importante para *smart grids*. Nessas redes, diferentes tipos de informações sensíveis estão sendo transmitidas entre diferentes entidades. Isso inclui informações privadas sobre usuários, como os tipos de dispositivos que eles possuem em casa, os períodos em que eles estão em casa, os lugares onde eles foram com seus carros, e informações sobre as concessionarias elétricas, esse último com aspectos econômicos importantes. Um erro comum é associar privacidade apenas com o uso de cifras. Na verdade, a criptografia é a principal maneira de proporcionar privacidade, mas só quando a autenticidade dos pontos finais de comunicação já foi comprovada.

- Integridade

  A integridade é a capacidade de garantir que os dados fluam dos remetentes para os receptores sem quaisquer alterações no conteúdo. Ataques *man-in-the middle* são usados para espionar as informações, e também para alterar o conteúdo das mensagens entre sua origem e destino(s). As comunicações em *smart grid* devem garantir a integridade, pois a modificação de dados transmitidos de ou para sensores ou atuadores podem causar interrupções na rede elétrica ou danos severos. Além disso, a comunicação entre os usuários e os sistemas deve ser protegida, a fim de evitar a má utilização do sistema que pode causar prejuízos financeiros para os usuários e perturbações na rede elétrica.

- Disponibilidade

  A disponibilidade está relacionada com falhas e ataques de negação de serviço (*Denial of Service* – DoS). Em um ambiente não hostil, falhas na rede, falhas no *hardware* e/ou *software*, ou uma sobrecarga de usuários em um servidor podem causar indisponibilidade do serviço. Em ambientes hostis, *hackers* podem usar um pequeno número ou um número elevado de dispositivos, normalmente controlados remotamente, para interromper um serviço. Esses ataques são chamados de DoS e DoS distribuído (DDoS - *Distributed DoS*), respectivamente. Esses tipos de ataques geralmente causam indisponibilidade do serviço e consequentemente perdas financeiras. Uma das principais preocupações sobre DoS ou DDoS é que eles são geralmente difíceis de parar sem prejudicar os usuários legítimos. A principal razão é que o tráfego gerado pelo atacante é semelhante ao tráfego legítimo e, portanto, sistemas de *firewall* não podem bloquear apenas o tráfego atacante. *Hackers* são capazes de gerar este tipo de tráfego proveniente de fontes distribuídas através de *botnets*. *Botnets* são compostas de um conjunto de dispositivos comprometidos por um código malicioso que pode ser operado remotamente. Normalmente, um usuário com um dispositivo comprometido não sabe que é parte de uma *botnet*, porque *bots* são geralmente transparentes para o usuário e geram pequenas quantidades de tráfego em momentos muito específicos, desencadeados por um usuário remoto mal intencionado. Este tipo de ataque é um dos principais motivos de preocupação em redes elétricas inteligentes, porque o sistema de energia é composto por um grande número de dispositivos que geram dados para serviços específicos, como o SCADA. Se um *hacker* comprometer os medidores inteligentes, poderá usá-los para interromper um serviço de coleta de informações de energia ou ainda comprometer sistemas de proteção e controle de *microgrids*.

- Proteção Física

  Para efetuar ataques, um *hacker* precisa acessar dispositivos. Em redes de energia legadas, dispositivos de controle eram fisicamente protegidos. Com isso, os *hackers* teriam de invadir fisicamente uma instalação da concessionária, a fim de acessar um dispositivo e perturbar a rede de controle. Com o avanço das redes elétricas inteligentes, a rede de controle está interligada ao usuário final através de dispositivos como medidores inteligentes. Assim, ao invés de tentar comprometer um dispositivo de controle ou serviço através da rede, procurando por vulnerabilidades de *software*, o *hacker* pode invadir a rede adulterando um medidor inteligente que está

em sua casa, por exemplo. Quando um *hacker* tem acesso físico a um nó, torna-se muito fácil a alteração de códigos e o acesso aos dados armazenados no dispositivo, mudando seu comportamento. Uma vez que um *hacker* controla um nó de rede legítimo, ele torna-se um atacante interno. Isto significa que o atacante controla um nó que tem a confiança de todo o sistema. Assim, todas as mensagens injetadas serão consideradas como legítimas. Após infectar um nó, torna-se mais fácil comprometer outros nós legítimos através da rede, porque os sistemas de segurança são configurados para bloquear as ameaças externas e liberar a comunicação entre nós internos. Assim, a segurança física de dispositivos de comunicação é uma das principais preocupações para as redes de energia.

4.3.1. Segurança em Tecnologia da Informação x Segurança em Sistemas de Controle Industriais

Uma das principais razões para a ocorrência de falhas de segurança do sistema de energia é a diferença entre cenários de segurança na Tecnologia da Informação (TI) tradicional e sistemas de controle industrial (ICS - *Industrial Control Systems*). Normalmente, para o provisionamento de segurança no mundo da Internet, confidencialidade é uma das questões principais, pois os dados do usuário não podem ser divulgados. No ICS, mesmo que a confidencialidade seja importante para proteger os segredos industriais, não é a principal preocupação. Integridade e disponibilidade são, de fato, os requisitos essenciais para a execução correta do sistema de controle, mesmo na presença de atacantes internos ou externos. Uma falha de privacidade pode causar prejuízos, mas uma falha causada por uma mensagem falsa ou uma mensagem que não chega pode destruir equipamentos e/ou causar danos ainda mais graves.

Nos sistemas tradicionais de TI, a regra principal é manter o sistema atualizado. *Patches* para resolver as questões de segurança devem ser aplicados o mais rápido possível para parar possíveis ataques usando a vulnerabilidade exposta. Nenhum engenheiro ou analista de sistemas teria medo de atualizar o sistema. Esta é uma realidade diferente no ICS. Na verdade, os dispositivos têm geralmente *firmware* proprietário, que pode falhar após uma atualização. Normalmente, os fabricantes de dispositivos não assumem a responsabilidade no *patch* do dispositivo e engenheiros não se sentem confortáveis para atualizar o *firmware* e correr o risco de comprometer dispositivos muito caros. Por isso, é comum substituir equipamentos por outros mais seguros em vez de atualizar o *software* do dispositivo (Lüders, 2011). Outra diferença importante é que, em sistemas de TI tradicionais, os dispositivos são nativamente integrados com listas de *firewall*, de controle de acesso (*Access Control List* - ACL), e outros sistemas de segurança, o que não é uma realidade no ICS. Além disso, computadores conectados à Internet podem contar com protocolos de comunicação seguros para compartilhar informações sensíveis, enquanto os dispositivos em um ICS, em geral, são baseados em protocolos de comunicação muito simples e sem preocupação com segurança.

Outra preocupação em ambientes ICS é que existe o hábito dos operadores de utilizar logins e senhas padrão. Assim, quando um atacante tem acesso à rede, geralmente, é muito fácil de acessar e controlar dispositivos diferentes.

Além dessas vulnerabilidades, o cenário dos ICSs tem outra particularidade: rara-

mente ocorrem janelas de manutenção no ICS, pois os dispositivos não podem parar sem causar prejuízos à cadeia produtiva. Por exemplo, para executar a manutenção em um disjuntor, esse dispositivo deve ser desativado. Em *datacenters* tradicionais, a manutenção da máquina não interfere na prestação de serviços, pois o uso da virtualização permite copiar e mover máquinas virtuais sem a interrupção de serviço. Por isso, não só é mais fácil aplicar *patches* em sistemas de TI tradicionais, mas também é mais simples e menos dispendioso abrir uma janela de manutenção. Além disso, o ICS tem de trabalhar com um grande número de dispositivos legados, o que aumenta o desafio de fornecer um ambiente de comunicação segura.

Por fim, os dispositivos dos ICS são desenvolvidos para atender casos de uso e não para os casos de abuso [Lüders 2011]. Logo, os fabricantes de dispositivos se concentram em funcionalidades ao invés de se concentrarem em robustez da rede. Em sistemas de TI, o *hardware* geralmente é para uso geral e há muitos esforços para proporcionar ao software robustez contra ataques cibernéticos. No ICS, o hardware é muito específico e com desenvolvimento fechado. Assim, em geral, apenas o fabricante pode desenvolver novos softwares. Portanto, a comunidade não é capaz de produzir *patches* tão rapidamente como acontece no mundo de TI. Além disso, na área de TI, há uma cultura de rápida disseminação de ameaças na Internet, enquanto no ICS há uma falta de vontade/hábito de pesquisar e compartilhar incidentes. O principal motivo é que os engenheiros dos ICS normalmente tentam resolver falhas rapidamente por meio de reinicializações do sistema ou pela substituição de um dispositivo danificado, em vez de tentar descobrir a origem do problema. Por isso, muitas vezes os ataques cibernéticos nem mesmo são identificados como um incidente de segurança cibernética, porque os engenheiros ainda não são capazes de diferenciar entre um ataque cibernético e uma falha de *hardware*/*software* [Wilhoit 2013]. Outra razão para não compartilhar dados de incidentes é que as empresas não querem espalhar suas vulnerabilidades. Consequentemente, descobrir e resolver vulnerabilidades torna-se muito difícil.

4.3.2. Ataques em Redes de Comunicação para *Smart Grids*

Esta seção descreve os cenários de redes inteligentes em que os ataques ocorrem devido a falhas de segurança de comunicação em rede. Primeiramente, os ataques contra subestação e cenários de supervisão são discutidos. Em seguida, são apresentados ataques contra a AMI.

4.3.2.1. Ataques contra Subestações e Centros de Controle de Dados

Antes de discutir os ataques, é necessário entender por que uma subestação ou um Centro de controle de Dados (*Data and Control Center* - DCC) pode representar um cenário vulnerável. Isso ocorre devido à arquitetura de comunicação e protocolos usados nesses cenários.

Um dos principais elementos de uma subestação é o sistema SCADA (*Supervisory Control and Data Acquisition*). O SCADA é utilizado não só para subestações, mas também para tipos diferentes de ICS. A implantação do SCADA não evoluiu muito nos

últimos 30 anos em termos de segurança da informação, apesar de vários problemas de segurança terem sido documentados (Wilhoit, 2013). No contexto das redes elétricas inteligentes, esta evolução é de especial preocupação, uma vez que redes de comunicação estão evoluindo para interligar o sistema como um todo, o que implica mais ameaças da rede. O SCADA controla e monitora remotamente os equipamentos da subestação a partir do DCC da concessionária, utilizando Unidades Terminais Remotas (*Remote Terminal Unit* - RTUs) localizadas em subestações e interconectadas através de uma rede de comunicação até o DCC. Mais recentemente, os IEDs são usados para a mesma funcionalidade.

O monitoramento é realizado através da aquisição de dados, tais como valores de correntes e tensões, e a notificação do status dos dispositivos de campo, como disjuntores. O controle está relacionado com a realização de comandos em dispositivos da subestação, como abertura e fechamento de disjuntores. Para isso é necessária uma rede de comunicação que interligue o SCADA (estação mestre) até o RTU (*Remote Terminal Unit*) (escravo), tal como ilustrado na Figura 4.4, e o uso dos chamados protocolos SCADA. Note que o IED pode comunicar-se diretamente com o SCADA utilizando algum protocolo específico, de modo que os RTUs poderiam ser removidos. Soluções com um grande número de dispositivos conectados a RTUs também são utilizadas, apesar do fato de que o controle remoto pode ser realizado diretamente no IED, como ilustrado na Figura 4.4.

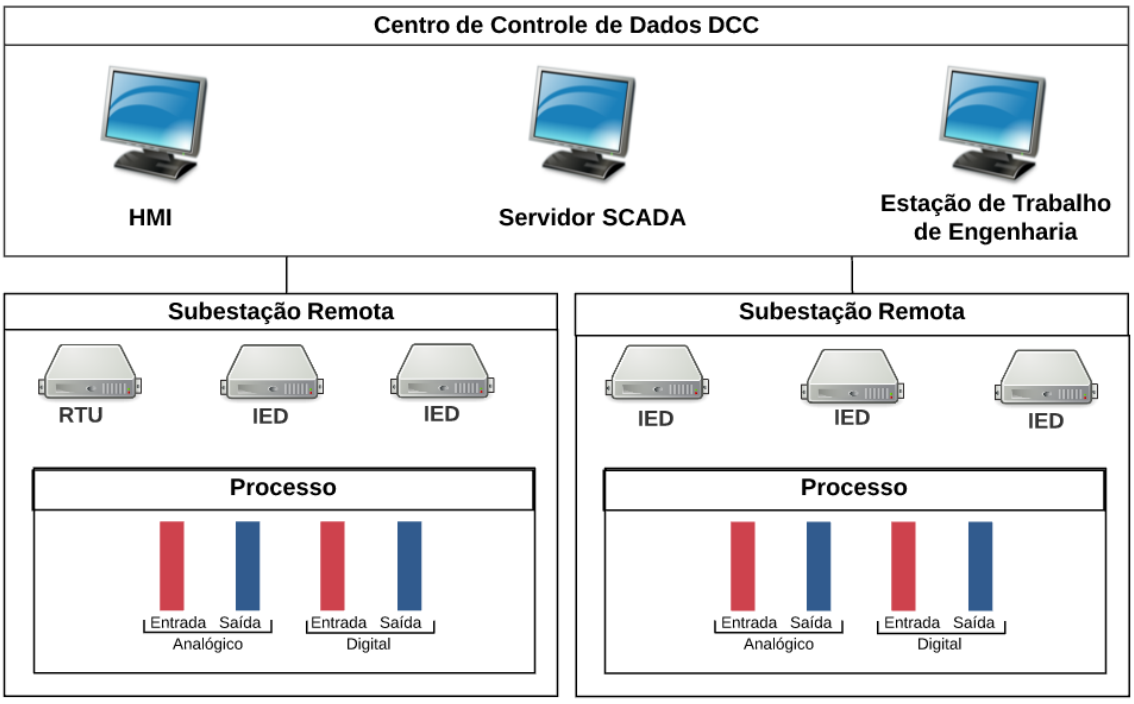


Figura 4.4. Esquema genérico de uma subestação.

A comunicação legada entre RTU e SCADA se dá por meio de protocolos SCADA e comunicação serial. Um dos protocolos SCADA mais antigos utilizados é MODBUS [Organization 2005], desenvolvido pela Modicon (atualmente Schneider) para sistemas de controle de processos industriais. O MODBUS, devido a necessidade de modernização, vem sendo atualizado desde 2005. No entanto, o DNP3 (*Distributed Network*

Protocol 3) [IEEE 2012] e o IEC 60870-5 [IEC 2007], ambos desenvolvidos na década de 90 e atualizados em 2012 e 2006 respectivamente, foram cada vez mais substituindo o protocolo MODBUS. Inicialmente, o DNP3 e o IEC 61870-5 foram criados para comunicação serial, como MODBUS. No entanto, em pouco tempo adquiriram versões para TCP/IP.

É importante enfatizar que a comunicação com esses protocolos acontece entre os RTUs/IEDs e o SCADA. Esta comunicação também é possível entre os IEDs e RTU. No entanto, nesse cenário o comando chega aos equipamentos através de cabos de controle conectados aos IEDs. Da mesma forma, as medidas de corrente e tensão, por exemplo, chegam ao IED através de fiação tradicional. Neste cenário, apenas a comunicação entre IED e SCADA é feita com comunicação e protocolos. Assim, um dispositivo de campo, tal como um disjuntor, recebe os comandos através de cabos de controle conectados aos IEDs (que podem ter recebido esse comando do SCADA).

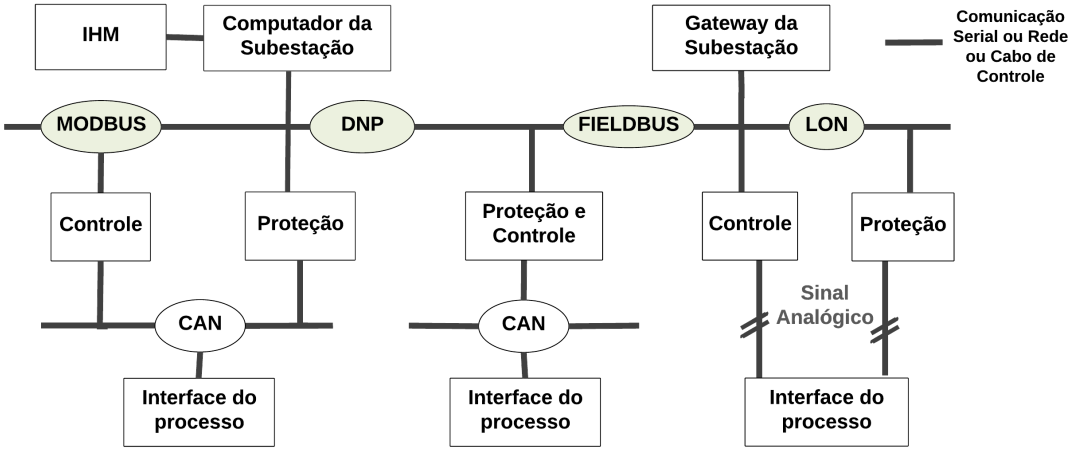
O DNP3, o IEC 60870-5, e os outros protocolos SCADA visam realizar o controle remoto e supervisão, mas não realizam funções de proteção elétrica. A filosofia de proteção, nesse cenário, é realizada sem o auxílio de protocolos e redes de comunicação. Por exemplo, depois de detectar uma condição anormal, o IED poderia iniciar um comando de proteção (trip) para abrir um disjuntor através de cabos de controle. Esse processo é feito automaticamente e de forma isolada, o mesmo IED que detectou o problema faz a tentativa de solução através de cabos de controle. Com isso, cada equipamento precisa estar diretamente conectado a um IED. Com a modernização das subestações, esses cabos de controle podem ser substituídos por uma rede de comunicação, como será discutido a seguir.

De fato, nos últimos anos, vários protocolos de automação de subestações que não são compatíveis uns com os outros foram propostos e implementados em subestações. A implantação de uma rede de subestação com diferentes protocolos trouxe muitos problemas para a automação de subestações, como dificuldade de manutenção, custos elevados, falta de interoperabilidade entre dispositivos de diferentes fabricantes, dentre outros. Para lidar com esses problemas, a norma IEC 61850 [IEC 2013] foi desenvolvida. Esta norma tem como objetivo garantir a interoperabilidade entre os dispositivos com o auxílio de uma modelagem própria e o uso de redes e sistemas de comunicação em subestações. Muitas concessionárias em todo o mundo já implantaram ou estão planejando implantar dispositivos de subestação baseados na norma IEC 61850 e redes de comunicação de acordo com esta norma [Budka et al. 2014]. A norma IEC 61850 define um modelo de objetos que representa formalmente as funções de proteção e controle, os equipamentos da subestação, a comunicação de dados e outros. Equipamentos de diferentes fornecedores podem ser instalados na mesma subestação desde que sejam implementados em uma rede de comunicação adequada e com os protocolos descritos na norma. Isso resulta em uma forte diferença entre a norma IEC 61850 e os protocolos SCADA tradicionais, tais como o DNP3 ou o IEC 60870-5, como ilustrado na Figura 4.5. A Figura 4.5(a) mostra o esquema de comunicação antes da implementação da norma IEC 61850, onde cada fornecedor usa um protocolo diferente para comunicar com o gateway da subestação e/ou com a IHM, seja a local ou a remota. Com isso, a implementação e a manutenção são mais caras além de mais difíceis e complexas. As equipes precisam de um conhecimento muito amplo para lidar com essa rede muito heterogênea. Além disso, protocolos como

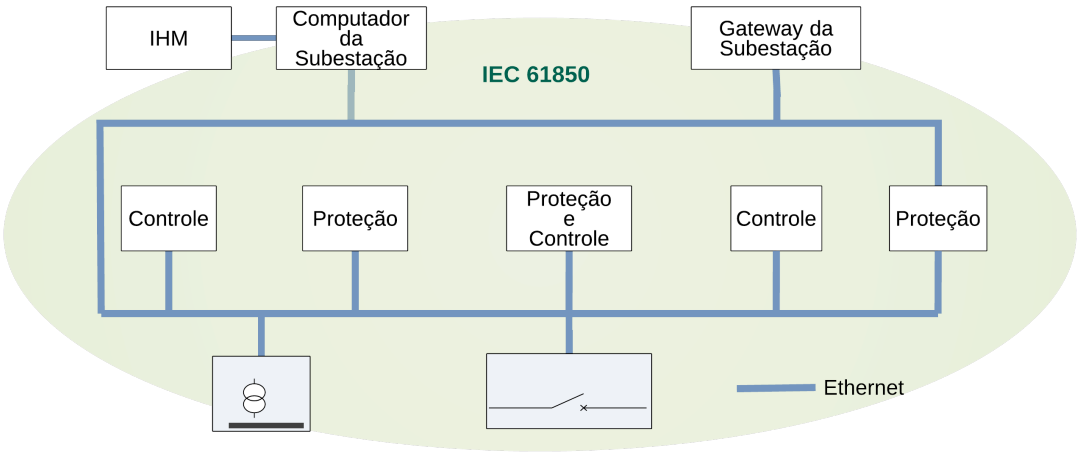
o DNP3 tem versões com comunicação serial (usando RS-232, RS-422 e/ou RS485) e rede ethernet, exigindo também o conhecimento de diversos padrões de camada física. Com a norma IEC 61850, como ilustrado na Figura 4.5(b), a comunicação é toda Ethernet de alta velocidade e padronizada com a modelagem IEC 61850. Nesse cenário, todos os equipamentos, independente de função ou de fornecedor, usam a mesma modelagem, facilitando a manutenção e configuração de toda rede. Além disso, com a norma IEC 61850, dispositivos de campo estão ligados por uma LAN Ethernet, substituindo os cabos de controle tradicionais. Portanto, os dispositivos de campo convencionais, como Transformadores de Corrente (TCs), Transformadores de Potencial (TPs), e disjuntores são substituídos por dispositivos modernos que se comunicam com os IEDs usando uma rede de comunicação e os protocolos padronizados na norma. A modelagem de dispositivos de automação é orientada a objeto e o modelo de comunicação utiliza três tipos de protocolos: GOOSE (*Generic Object Oriented Substation Event*), SV (*Sampled Values*) e MMS (*Manufacturing Message Specification*).

O MMS é um protocolo SCADA que é muito semelhante ao DNP3. Este protocolo utiliza um modelo cliente-servidor, onde os IEDs são servidores e o cliente é SCADA. O MMS usa as sete camadas do modelo OSI e seu atraso varia de 100 ms a 1000 ms. O protocolo GOOSE e o SV têm objetivos diferentes dos protocolos SCADA, e com isso um comportamento bastante diferente. Ambos são usados em esquemas de proteção e controle, e com isso têm restrições temporais muito mais rígidas, já que são usados em esquemas automáticos sem interação humana. Estes protocolos usam o modelo *publish-subscribe* (publicador-assinante) com um endereçamento MAC (*Media Access Control*) multicast. O SV também pode usar o modelo cliente-servidor e endereços unicast. Ambos têm limitações de tempo de até 3 ms e estão diretamente mapeados na camada de enlace, a fim de fornecer um tempo de resposta mais rápido. Essa restrição temporal rígida ocorre pois as mensagens GOOSE e SV são usadas em esquemas de proteção da rede elétrica. A mensagem SV é usada para enviar medidas de transformadores de instrumento e/ ou *Merging Units* e a mensagem GOOSE é usada para os esquemas de proteção e automatismos. Assim, o protocolo GOOSE e o SV permitem a comunicação entre os dispositivos da subestação e não envolvem o SCADA. Por exemplo, TCs e TPs podem enviar medições através de mensagens SV para os IEDs. Depois de detectar uma condição anormal analisando as mensagens SVs recebidas, o IED pode iniciar um comando para abrir um disjuntor (trip). No entanto, se este disjuntor falhar, o IED pode enviar uma mensagem GOOSE indicando que houve uma falha na abertura do seu disjuntor (*breaker failure*) para outros IEDs como um esforço para resolver o problema de outra forma e o mais rápido possível.

Muitas aplicações de energia em *smart grids* têm limitações de tempo rígidas em termos de disponibilidade de comunicação e atraso [IEC 2009]. Portanto, características específicas deste novo conceito de entrega de energia têm impulsionado vários projetos de pesquisa que visam a concepção de uma infraestrutura de comunicação adequada para atender a qualidade de serviço (QoS - *Quality of Service*) e a confiabilidade esperada para as redes elétricas inteligentes [Kounev et al. 2016]. Por exemplo, a norma IEC 61850 abordou o problema da inserção de recursos de energia distribuídos no sistema (DER - *Distributed Energy Resources*) [IEC 2009], recomendando o mesmo limite de tempo estabelecido para a proteção e controle em subestações.



(a) Esquema de comunicação de subestações legadas.



(b) Esquema de comunicação usando IEC61850

Figura 4.5. Comparação entre IEC61850 e outros esquemas de comunicação [Lopes et al. 2012].

A norma IEC 61850 recomenda atrasos de 3 ms a 100 ms para mensagens de proteção de acordo com o tipo de mensagem. Além disso, em 2010, o Departamento de Energia dos Estados Unidos analisou os requisitos de comunicação para funções das redes elétricas inteligentes (por exemplo, resposta à demanda e DER) e definiu valores da ordem de milissegundos para a proteção e controle de *smart grids* além de perfis de confiabilidade para cada serviço [DoE 2010]. Restrições temporais rígidas também foram descritas pela norma IEEE 1646 [1646 2004]. A norma IEEE 1646 firma requisitos de atraso para algumas operações de subestações em 4 ms e 5 ms, para frequências AC de 60 Hz e 50 Hz respectivamente.

Para as aplicações que requerem comunicação entre subestações, os requisitos de atraso são mais permissivos. Assim, a ativação de um esquema de proteção em uma

subestação deve ser iniciado em 8 ms após uma falha ser detectada numa subestação adjacente. Como consequência deste novo padrão de comunicação, a utilização de IEDs em subestações resultou em muitas vantagens tais como a comunicação de alta velocidade e custos reduzidos. No entanto, as melhorias deste sistema digital geram várias ameaças de segurança em subestações. Os ataques podem alterar os dados sendo enviados para esta rede, o que pode causar, por exemplo, uma abertura ou fechamento indevido de disjuntores, como discutido antes. No caso da abertura indevida, o sistema deixa de fornecer energia para as cargas sem que haja qualquer falha, causando uma interrupção desnecessária no fornecimento de energia para o consumidor. No caso de um fechamento indevido, mesmo em condição de falha o sistema é restabelecido, configurando um curto-circuito. Além disso, se o circuito estiver em manutenção, um fechamento indevido pode ameaçar a vida humana.

As próximas seções apresentam a descrição de ataques que podem causar grandes danos na subestação. Os ataques são subdivididos em dois tipos: o tipo um representa os ataques contra o SCADA; o tipo dois representa os ataques que podem ser executados após os ataques contra o SCADA, quando o atacante já está localmente na subestação.

4.3.2.2. Tipo 1: Ataques contra os sistemas de supervisão

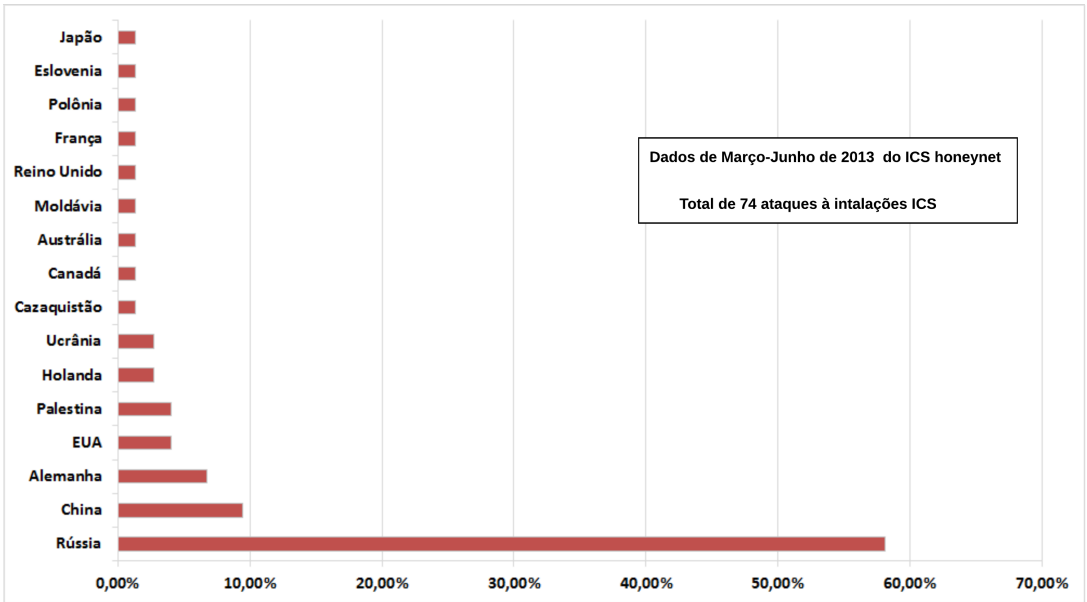


Figura 4.6. Origem dos ataques contra alvos ICS [Wilhoit 2013].

Os ataques contra os sistemas de supervisão acontecem em qualquer tipo de ICS. Um estudo recente espalhou uma série de *honeypots* em todo o mundo emulando um ICS operando com um SCADA e os protocolos de comunicação MODBUS e DNP3 controlando um sistema de bomba. Um *honeypot* é uma instalação que cria uma versão totalmente mimetizada de uma instalação real. A ideia é criar um ambiente atraente para os *hackers*,

a fim de estudar novas formas de ataques [Wilhoit 2013]. Esse estudo registrou 74 ataques específicos às instalações do ICS em um período de três meses. O número de tentativas de ataque foi ainda maior se considerarmos ataques automatizados genéricos como injeções SQL, atingindo 33.466 ataques. A Figura 4.6 mostra a distribuição da origem de ataques específicos a ICS.

Os ataques observados estavam relacionados a vulnerabilidades do SNMP (*Simple Network Management Protocol*), do servidor da IHM, da ausência de um sistema de autenticação adequado e de um VxWorks (*File Transfer Protocol* - FTP). Para melhor entender os ataques contra o SCADA, é preciso estudar a utilização do protocolo DNP3. Outros protocolos de comunicação, tais como MODBUS ou MMS no IEC 61850 sofrem ataques semelhantes, já que nenhum destes protocolos foi projetado considerando-se a existência de um ambiente de comunicação não-confiável. Assim, estes protocolos não empregam de forma nativa criptografia, autenticação e autorização.

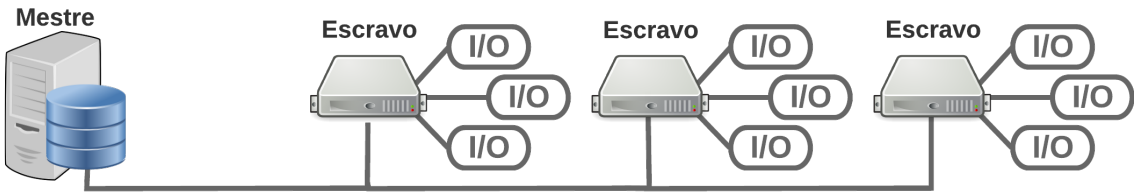
Em geral, os ataques contra os sistemas que usam o SCADA são divididos em três categorias: ataques que exploram especificações do protocolo de comunicação; ataques que exploram implementações do fabricante, como erros de configuração e falhas de código; e os ataques contra a infraestrutura subjacente, que têm como alvo a tecnologia da informação, os ativos de rede, e as políticas de segurança fracas do sistema [East et al. 2009]. Como os ataques aos protocolos SCADA são similares, nesse capítulo o DNP3 foi escolhido como enfoque.

O DNP3 permite três topologias possíveis entre o mestre e o dispositivo escravo (outstation) a ponto-a-ponto, a ponto-multiponto e a hierárquica, como mostrado na Figura 4.7. A comunicação entre o mestre e os dispositivos escravos é modelada de três maneiras diferentes: *unicast*, *broadcast*, e respostas não solicitadas. No modo unicast, o mestre envia uma solicitação e aguarda por uma resposta do escravo. Por exemplo, o mestre pode solicitar o estado do disjuntor ou executar um comando no disjuntor e o escravo responde com a leitura solicitada ou com o resultado da operação comandada, respectivamente. No *broadcast*, um pedido é encaminhado para todos os dispositivos escravos e não há resposta para o mestre. Na resposta não solicitada, dispositivos escravos enviam uma mensagem não solicitada para o mestre contendo atualizações periódicas, eventos ou alertas.

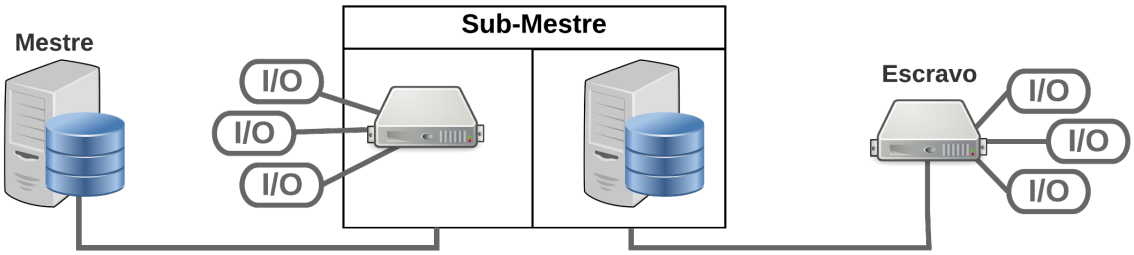
Ataques direcionados/provenientes do sistema de supervisão são baseados em interceptação de mensagens, injeção de mensagens falsas, e modificação de mensagens. Ataques contra redes trafegando o protocolo DNP3 podem ser classificados de acordo com a camada de arquitetura de rede onde ele ocorre. Seguem alguns exemplos de ataques contra o DNP3 [East et al. 2009]:

- Reconhecimento passivo de rede: O atacante com acesso adequado captura e analisa as mensagens que trafegam na rede para descobrir informações sobre a topologia de rede, os dispositivos em uso, as funcionalidades disponíveis, etc.
- Repetição de resposta *baseline* e *man-in-the-middle*: Nesses ataques, um invasor observa o tráfego de rede e injeta mensagens para o mestre passando-se por um dispositivo escravo e vice versa: para dispositivos escravos se passando pelo mestre. No caso do ataque *man-in-the-middle*, um dispositivo é colocado entre o mestre e

(a) Topologia ponto a ponto



(b) Topologia ponto-multiponto



(c) Topologia hierárquica

Figura 4.7. Topologias para DNP3 [East et al. 2009].

os escravos capturando e modificando o tráfego e personificando o outro. Os objetivos destes ataques são: espionar as informações que trafegam na rede, interromper o funcionamento do mestre e/ou dos escravos, modificar o comportamento do mestre e/ou dos escravos, e representar o mestre e/ou escravos para causar falhas no serviço.

- Modificação de campos da camada de enlace: Este ataque, que depende do estabelecimento de um ataque *man-in-the-middle*, tem muitas variações, de acordo com o campo de mensagem DNP3 que é modificado. O formato do quadro DNP3 é descrito na Figura 4.8. Por exemplo, o atacante poderia modificar o campo comprimento para interromper o processamento de mensagens; mudar o flag DFC para enviar um sinal falso de escravo ocupado para o mestre; ou mudar a mensagem para enviar o Código de Função 1, a fim de promover uma reinicialização desnecessária do escravo causando uma indisponibilidade temporária.

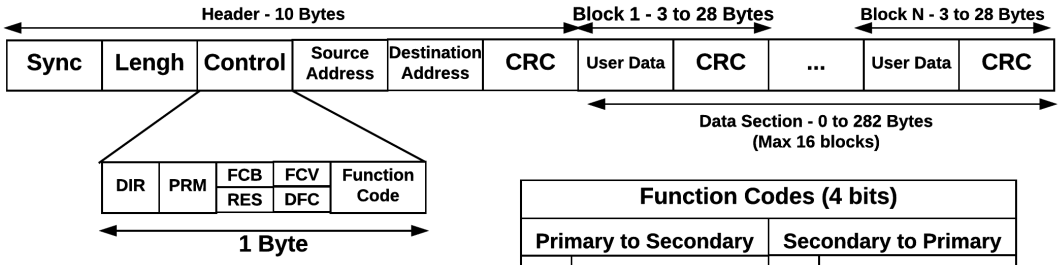
- Modificação de campos da pseudo camada de transporte: A chamada pseudo camada de transporte do DNP3 cuida da fragmentação dos pacotes. Este ataque é uma outra variação do *man-in-the-middle* para interromper o tratamento de mensagens fragmentadas. Neste caso, o atacante poderia mudar campos da mensagem de

transporte fazendo com que o destino descarte todos os fragmentos incompletos ou ainda causando erros de processamento ao juntar informações fragmentadas.

- Ataque de comandos em escravos: Nesta aplicação de ataque, o atacante usa um comando falso para gravar dados falsos em um escravo. Este ataque envia uma mensagem DNP3 com um *function code* (FC) que escreve objetos de dados falsos em um escravo, causando erros no dispositivo. Outra variação deste ataque é o uso de outros FCs para congelar e limpar os objetos de dados já existentes nos escravos, criando estados inconsistentes no sistema.

- Interceptação de arquivo de configuração: Esta aplicação de ataque visa a obtenção do arquivo de configuração de um escravo. Para fazer isso, o invasor envia uma mensagem indicando um arquivo de configuração corrompido enquanto representa a identidade do mestre. A estação escrava vítima, em seguida, reenvia o arquivo de configuração, que é interceptado pelo atacante.

- Negação de serviço com um único pacote: Neste ataque, o invasor envia pacotes de resposta especialmente montados que são capazes de travar o mestre. Este ataque explora ambos *firmware* e DNP3, visando interromper todo o sistema da subestação, uma vez que é capaz de parar o mestre. Como consequência, o centro de controle não pode mais monitorar e controlar a rede do SCADA. O ataque pode ser desencadeado por um pedido do mestre ou por qualquer outro evento escolhido pelo atacante, já que o DNP3 permite também que sejam enviadas respostas não solicitadas pelos escravos.



Function Codes (4 bits)

Primary to Secondary		Secondary to Primary	
Code	Description	Code	Description
0	Reset link	0	ACK
1	Reset user process	1	NACK
2	Test link	11	Response - link status
3	User data- require ack	14	Link down
4	User data - no ack	15	Link not used
9	Request link status		

Figura 4.8. Formato do quadro DNP3.

É importante notar que esses ataques são documentados e podem ser facilmente realizados utilizando ferramentas abertas [Rodofile et al. 2015]. A principal dificuldade

é obter o acesso à rede executando o sistema SCADA, que é pra ser uma rede segura. Os centro de controle de empresas, onde ficam os sistemas SCADA, costumam ter a rede complemente isolada da rede corporativa. Além disso, os dispositivos atacados estão fisicamente isolados dentro de subestações. Isso faz com que a preocupação com a segurança seja minimizada. Apesar disso, alguns ataques já foram relatados nesse tipo de rede mostrando que mesmo isoladas estão sujeitas a ataques [Wei and Wang 2014]. Com a implementação das redes elétrica inteligentes a proteção física deixa de existir fazendo com que as ameaças aumentem consideravelmente, o que requer uma redefinição de protocolos de comunicação para incluir segurança, considerando que a rede não está mais isolada em um ambiente de *smart grid*.

4.3.2.3. Tipo 2: Ataques contra a comunicação de dispositivos locais

Ataques contra os dispositivos locais da subestação dedicam-se ao uso indevido de um IED para perturbar o sistema elétrico. Detalhes específicos desses ataques dependem diretamente do protocolo de comunicação em uso. Para ilustrar, os protocolos da norma IEC 61850 serão usados como exemplo, já que além de um protocolo SCADA (MMS), a norma possui protocolos para realização de proteções na subestação (GOOSE e SV). Tanto o acesso remoto não autorizado quanto o acesso físico não autorizado podem resultar nos ataques descritos a seguir.

É importante perceber que há diferentes métodos para um invasor acessar um IED. A maneira mais simples é quando um atacante interno acessa o dispositivo e muda os parâmetros de configuração para danificar a rede. Outra possível ação de um invasor interno é conectar um dispositivo mal intencionado na rede, a fim de injetar um tráfego adulterado personificando os dispositivos desta rede. No entanto, também é possível ter acesso a um IED através de métodos externos, através de *exploits*, ou atacando um dispositivo que liga a subestação com o mundo exterior. Outra possibilidade é acessar um IED através do SCADA. Uma vez que o SCADA é comprometido, torna-se muito fácil o acesso aos IEDs, porque geralmente esses dispositivos são configurados com um login e senha padrão.

Uma das principais preocupações quando se analisa a comunicação dentro de uma subestação é que os requisitos de QoS de mensagens de proteção não são compatíveis com atrasos impostos pelos métodos de criptografia. Para proporcionar autenticidade e integridade, que são os requisitos mais básicos de segurança em sistemas de controle, é necessário que algum esquema de criptografia básico seja realizado. Um grande número de ataques tornam-se possíveis nesse cenário, já que não há nenhuma autenticação ou verificação de integridade nos protocolos de comunicação atuais, além de todas as outras vulnerabilidades do ICS descritas nas seções anteriores.

Esta seção concentra-se em ataques contra redes IEC 61850, a fim de ilustrar os impactos dos ataques realizados na comunicação entre IEDs e também entre todos os dispositivos locais. O protocolo GOOSE terá maior enfoque para exemplificar os ataques, pois permite a comunicação entre os IEDs. O foco principal do protocolo GOOSE é a transmissão de dados rápida e confiável entre dois ou mais IEDs. Mesmo assim, quando se utiliza GOOSE, uma subestação é propensa a diversos ataques, tais como:

- Ataque de negação de serviço: Este ataque é usado para impedir que usuários acessem recursos de rede. O atacante envia um grande número de mensagens para a máquina sob ataque usando uma ou mais máquinas já comprometidas. No cenário de subestação, este ataque visa parar um IED ou um concentrador local da subestação (no caso do uso de MMS). Além disso, o atacante provavelmente tem a intenção de retardar a entrega de mensagens críticas, como GOOSE e SV, entre as subestações e/ou desativar funções de monitoração e controle remoto, que usem MMS [Bayat et al. 2015]. Danos sérios podem ocorrer em subestações, uma vez que a comunicação é invadida e que o atacante impede a recepção de tráfego legítimo. Para executar este ataque, o atacante pode acessar o IED utilizando *exploits* de *firmware* ou contornando as medidas de segurança de rede. Uma vez que o atacante controla um IED, ele gera uma enorme quantidade de pacotes GOOSE para a rede da subestação. Como mensagens GOOSE são enviadas como se fossem em broadcast, todos os dispositivos da subestação começam a receber um grande número de mensagens GOOSE. Este ataque é também chamado de ataque de *flooding* [Li et al. 2015]. Duas consequências surgem: as mensagens legítimas podem não chegar ao destino em tempo por causa de filas de mensagens em *switches* de rede e em terminais; os IEDs podem parar de funcionar porque eles não são projetados para receber esse excesso de mensagens. Esta segunda consequência é mais fácil de observar se o atacante usa mensagens mal formadas [Khaitan et al. 2015, Lopes et al. 2015b, Noce et al. 2016].

- Falsificação de GOOSE (*spoofing*): Já que não há nenhuma autenticação ou verificação de integridade em mensagens GOOSE, os atacantes são capazes de enviar mensagens falsas na rede. Para injetar tráfego consistente, um atacante pode observar o tráfego de rede para descobrir dados como o número de status atual (stNum) de um fluxo de mensagens GOOSE. O parâmetro stNum funciona como um número de sequência. Assim, o atacante pode gerar mensagens GOOSE incrementando o stNum depois de inspecionar uma mensagem GOOSE inicial com o stNum verdadeiro. As mensagens GOOSE falsas são enviadas em *multicast* o mais rapidamente possível pelo atacante, com um número de stNum maior que o verdadeiro. Uma vez que o tráfego de ataque começa a ser processado pelo assinante, o tráfego legítimo com números de stNum mais baixos serão descartados [Kush et al. 2014]. Portanto, o atacante para o fluxo de informação legítimo, além de poder inserir qualquer tipo de informação falsa que possa afetar a rede de comunicação ou o sistema de potência.

- Personificação do dispositivo central: Neste ataque, o dispositivo atacante falsifica a identidade de um servidor do sistema de supervisão. É mais fácil de ser implantado se o atacante é capaz de conectar um computador à LAN da subestação. Um software de automação industrial que permita aos clientes implementar um SCADA pode ser usado para esse ataque, e, de fato, esses tipos de softwares estão facilmente disponíveis. Uma vez que o software estabeleça comunicação com um IED como mestre, qualquer comando pode ser executado prejudicando a subestação.

- Ataques contra Ethernet: O protocolo GOOSE especifica o uso de Ethernet para conectar dispositivos na LAN da subestação. Portanto, esta rede é propensa a todos

os ataques de camada 2 contra Ethernet, tais como ataques contra o ARP, ataques de inundação de MAC, ataques contra a *Spanning-Tree*, ataques de força bruta contra o *multicast*, ataques contra o VLAN *trunking*, ataques contra VLANs privadas, roubo de identidade, etc [Yoo and Shon 2015].

4.3.3. Ataques à Infrastrutura de Medição Avançada

À medida que a complexidade e o grau de automação nas plantas industriais e na infraestrutura das companhias de energia elétrica aumentaram, a necessidade de um sistema confiável e flexível que poderia permitir a coleta de medições em localizações geográficas afastadas ou lugares perigosos, levou a indústria a desenvolver uma infraestrutura de dispositivos com capacidades de processamento e de telecomunicações. Estes dispositivos são conhecidos como medidores inteligentes.

Infrastrutura de Medição Avançada (AMI) é um sistema de comando e controle que tem milhões de nós e atinge todos os consumidores e quase todos os sistemas da empresa. Com a utilização de medidores inteligentes, que coletam grandes quantidades de dados, e com a implantação do AMI, a necessidade de segurança na distribuição de energia torna-se evidente. Nesta seção, os tipos de ataques contra AMI serão apresentados e também as ameaças e as vulnerabilidades na rede de acesso.

4.3.3.1. Visão Geral da Infrastrutura de Medição Avançada

A implantação de uma comunicação bidirecional é o elemento chave de *smart grids*. Da mesma forma, a introdução de medidores inteligentes na rede de distribuição permite uma melhor compreensão da demanda e um melhor controle do consumo de energia e geração distribuída. A infraestrutura de medição avançada é uma parte essencial de um sistema de distribuição inteligente e refere-se à rede que conecta o operador de distribuição com o cliente. No final do operador, um sistema conhecido como sistema de gerenciamento de dados do medidor (*Meter Data and Management System* - MDMS) interliga medidores eletrônicos capazes de coletar informações precisas com base no tempo sobre o consumo de energia dos clientes.

Abordagens comuns para as redes de medidores são a ligação direta com o MDMS dentro do centro de dados e controle (*Data and Control Center* - DCC) ou através de um concentrador de medidores, como mostra a Figura 4.9. Esta rede local de medidores que se comunicam com um concentrador é conhecida como NAN (*Neighborhood Area Network*). Tecnologias popularmente usadas nessa comunicação são RF Mesh ou comunicação via rede elétrica nas frequências de banda estreita (*Power Line Communications over narrowband frequencies* - PLC-NB). PLC limita o número de medidores ligados a dispositivos nos enrolamentos secundários do transformador em que o concentrador é instalado. Assim, PLC é geralmente menos empregado do que uma alternativa de comunicação sem fio [Budka et al. 2014].

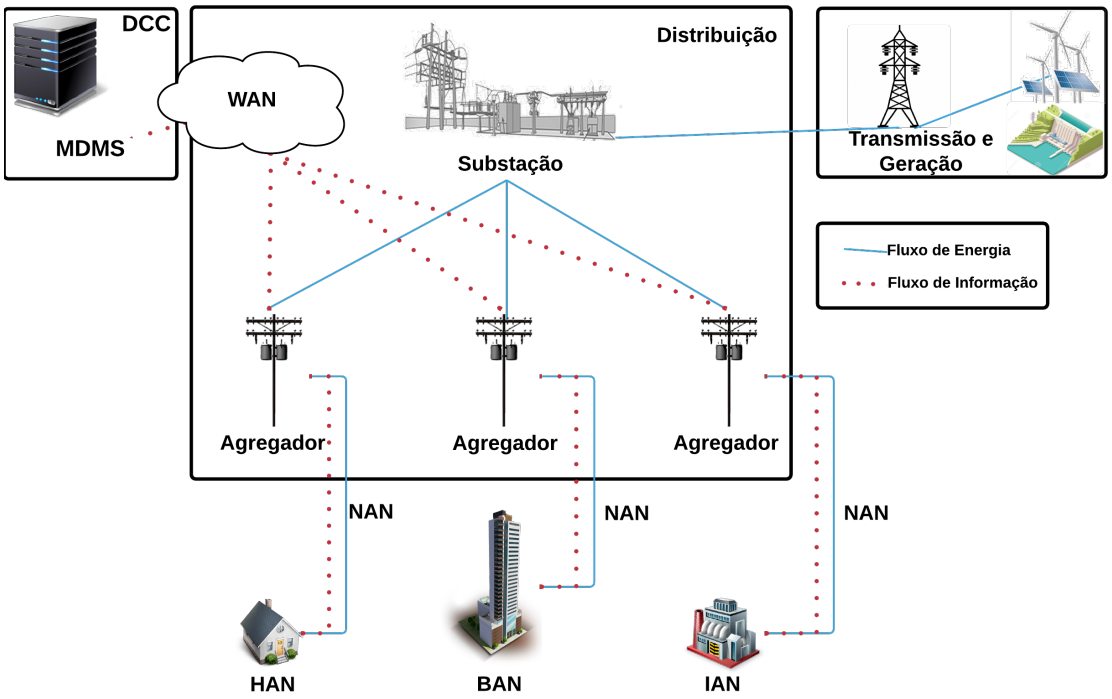


Figura 4.9. Estrutura AMI no contexto de *smart grid*.

Algumas das novas funcionalidades introduzidas com a implantação de medidores inteligentes, tais como informações de preços dinâmicos e a alta precisão e medição em tempo real do consumo de energia do consumidor, inserem uma série de vulnerabilidades que podem expor a privacidade do usuário. Essas vulnerabilidades são devidas à precisão da informação gerada por estes medidores. A assinatura elétrica de muitos aparelhos domésticos e atividade humana podem ser rastreadas por um invasor. Dados privados podem ser usados para roubo, sequestro e outras atividades criminosas. Informações sobre preços encorajam os consumidores a evitar o consumo de energia em horas de pico de demanda, e a controlar o seu consumo de energia de uma forma mais consciente, mas esses preços também podem ser manipulados para se controlar o mercado de energia. Ao confiar em tecnologias sem fio, uma NAN se torna vulnerável a sinal de interferência, espionagem, ataque de repetição, e ataques de injeção de dados. Estes são alguns exemplos da importância de investir em um sistema de comunicação seguro para a AMI que serão detalhados nas seções seguintes [Finster and Baumgart 2015].

Nas seções seguintes, este capítulo descreve ataques em estruturas cibernéticas internas da rede de distribuição. Os ataques contra a infraestrutura de medição avançada e à HAN (*Home Area Network*), tais como ataques contra a privacidade do usuário, os ataques contra o serviço de distribuição, bloqueio de sinalização e outro uso malicioso da rede de comunicação. De acordo com o relatório NIST sobre segurança cibernética para *smart grid*, como discutido antes, três principais objetivos para uma rede segura são: disponibilidade, integridade e confidencialidade [Group 2010]. No contexto de automação de distribuição e AMI, estes conceitos são aplicáveis como se segue:

- Disponibilidade: O acesso a funcionalidades do sistema deve estar pronto quando necessário. Se um ataque interrompe a comunicação entre uma casa inteligente e o centro da operação, ele compromete a disponibilidade do sistema.

- Integridade: A informação deve ser protegida contra a falsificação, alteração ou destruição. No contexto de NANs, um exemplo de perda de integridade é a modificação das informações de consumo de energia por um cliente malicioso que tenta negar sua responsabilidade financeira.

- Confidencialidade: O acesso à informação deve ser restrito a entidades autorizadas, a fim de proteger a privacidade e informações confidenciais. Esta é uma grande preocupação para os clientes, uma vez que um atacante pode adquirir uma grande quantidade de informações pessoais de um sistema preciso de monitoramento de energia.

4.3.3.2. Ataques contra a Disponibilidade de Serviço em Sistemas de Distribuição

Esta seção apresenta exemplos de ataques contra disponibilidade no sistema de distribuição e seus impactos. Ataques à disponibilidade tentam interromper a operação normal dos serviços e podem ser realizados em diferentes camadas de comunicação. Conforme os protocolos de comunicação para NANs forem escolhidos, outras vulnerabilidades podem surgir. Aqui vamos nos concentrar no bloqueio do canal, um ataque simples e genérico na camada física. O bloqueio consiste na transmissão de um sinal de interferência que diminui a relação sinal-ruído de um canal de comunicação sem fio.

A manutenção do equilíbrio entre a produção e consumo de energia é essencial para a estabilidade da rede. Com *smart grid*, a introdução de fontes de energia renováveis aumentou. Assim, a predição da energia produzida torna-se mais difícil, devido à natureza intermitente das fontes renováveis. Fontes de energia renováveis dependem de fatores ambientais que tornam a previsão de geração de energia mais complexa e menos precisa. Portanto, existe uma mudança de paradigma com a modernização de rede elétrica: na rede tradicional, a produção adapta-se à demanda, mas nas redes elétricas inteligentes, a demanda adapta-se à produção e faz com que o consumo de usuário seja mais eficiente. Os programas DSM (*Demand Side Management*) surgem como uma das soluções para ajustar o consumo do usuário à geração. DSM é uma ação ou decisão tomada pela empresa de energia para alterar ou modelar o padrão de consumo do usuário. O funcionamento correto da DSM depende de uma comunicação confiável entre a operadora e os consumidores. Dois exemplos de ataques de bloqueio de sinal contra programas DSM são discutidos a seguir, com diferentes motivações, que podem resultar em queda de energia.

No contexto de resposta à demanda em tempo real, um primeiro exemplo de ataque é o descrito a seguir. Nos programas em tempo real, o preço da energia é dinâmico ao longo do dia. O mercado usa a demanda de energia, o custo de geração de energia e as restrições das linhas de transmissão para calcular o preço que reflete a disponibilidade de recursos na rede. Em seguida, os usuários deste programa recebem mensagens do mercado a cada hora que custos de energia muda e ajustam seu consumo ao novo

preço. Li e Han descrevem uma possibilidade de manipulação do mercado por interferência do sinal de preço entre o mercado e os consumidores, como mostra a Figura 4.10 [Li and Han 2011]. Quando há baixa disponibilidade de energia, o mercado envia uma mensagem com um preço mais elevado para que os usuários reduzam o seu consumo e esperem por uma mensagem de preço mais baixo para aumentar ou normalizar o consumo. O atacante bloqueia o sinal de preço de uma área densamente povoada, enquanto os sistemas dos consumidores continuam trabalhando com o último preço recebido, e o atacante monitora o preço do mercado à espera de uma mudança significativa para parar a interceptação do sinal. Portanto, o atacante pode controlar as alterações de preços e usá-las para o lucro, por exemplo, se o sinal é bloqueado durante um preço mais elevado, quando o preço diminui, ele armazena energia, enquanto os outros usuários estão trabalhando com um preço mais elevado. Em seguida ele para o bloqueio, os usuários irão receber um preço mais baixo e vão aumentar o seu consumo, o preço tenderá a aumentar de novo e, neste momento, ele vende a energia armazenada. Como a operadora usa o preço para equilibrar a demanda e a oferta, se uma área densamente povoada não receber um aumento de preços e não reduzir o seu consumo, pode ocorrer um instabilidade na rede ou mesmo o apagão de uma grande área.

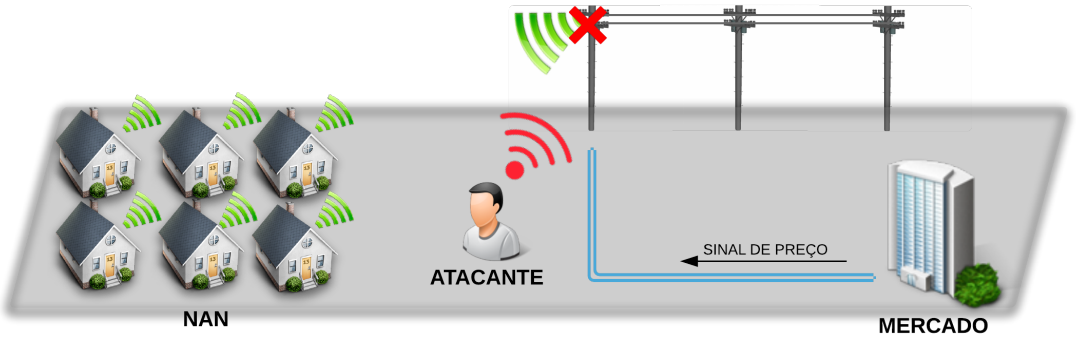


Figura 4.10. Bloqueio do sinal de preço para manipular o mercado de energia.

O segundo exemplo ocorre no controle direto de carga. DLC (*Direct load control*) é uma alternativa de programa DSM, em que o consumidor recebe incentivos ou descontos na conta de energia para permitir que a operadora tenha o controle direto de alguns aparelhos de sua casa. Em casos de emergência ou quando a demanda excede a oferta disponível, uma mensagem de comando é enviada para desligar algumas cargas e proteger a rede. Comandos de controle para reduzir o consumo de energia passam através da rede AMI, e semelhante aos programas de preços em tempo real, a operadora usa o DLC para equilibrar a demanda e a oferta, então, a intercepção de um comando de reduzir pode afetar a estabilidade da rede. Ataques contra esta comunicação podem danificar a rede. O impacto pode ser desde um simples desconforto dos usuários ou até mesmo a falta de energia em áreas críticas, comprometendo vidas humanas. Um usuário mal intencionado pode interromper esta comunicação e bloquear mensagens e assim enganar o programa, para que ele receba desconto na fatura por participar do programa sem desligar qualquer

carga. Após um caso de emergência, a operadora envia um comando para normalizar o consumo do usuário. Um adversário pode bloquear a chegada da mensagem impedindo a normalização de operação de um usuário específico. No pior dos casos, ataques terroristas podem bloquear o sinal em uma grande área, afetar a estabilidade da rede e causar falta de energia.

4.3.3.3. Ataques contra a Integridade dos Dados em Sistemas Distribuídos

Ataques que tentam manipular dados ao invés de bloquear serviços são tipicamente mais sofisticados do que um ataque de bloqueio. Além disso, as suas consequências são geralmente mais graves. Um atacante pode modificar dados para fraudar informações de consumo de energia em nome de um cliente. Outra possibilidade envolve a emissão de um comando de interrupção do serviço para um medidor, deixando uma residência sem energia. Além disso, o *firmware* do medidor é vulnerável a injeção de *malware* durante a atualização, ou um *firmware* modificado pode ser carregado comprometendo a sua capacidade de faturamento. Outra possibilidade é a de comprometer um grande número de dispositivos para corromper a visão global do sistema de detecção. Neste caso, um grande número de dispositivos seria capaz de injetar mensagens falsas na rede, que contém dados de medição falsa ou alarmes falsos. Uma falsa visão global do sistema poderia desencadear ações erradas através do sistema de supervisão de distribuição, causando falhas de energia.

A metodologia de ataque é muito semelhante aos ataques contra a Internet. Por exemplo, um invasor pode executar um *man-in-the-middle*, executar um ataque de repetição ou até mesmo criar um *botnet* de medidores inteligentes. No caso de um ataque *man-in-the-middle*, o usuário mal intencionado vai tentar personificar um medidor inteligente de confiança e/ou o DCC. O atacante vai se comunicar personificando os pares finais. Essa manobra permite a um atacante espionar dados, injetar falsos pacotes, reenviar os dados autênticos, e modificar a carga útil dos pacotes [Ur-Rehman et al. 2015]. Ataques de repetição são mais simples de executar, uma vez que eles são baseados em reenviar mensagens antigas, mas eles também são mais fáceis de evitar. Injetar novas mensagens falsas se passando por um medidor seria muito mais eficaz do que um ataque de repetição. Por exemplo, um usuário malicioso pode roubar as credenciais de um medidor inteligente fisicamente corrompendo o dispositivo. Em seguida, o usuário mal intencionado envia dados falsos usando as credenciais válidas de um computador. Normalmente dispositivos à prova de falsificação que evitem roubos de credenciais são caros e não se espera que os medidores inteligentes atendam este requisito. A última metodologia de ataque seria invadir um ou mais medidores através da rede de comunicação. Corrompendo muitos medidores, o *hacker* poderia criar um *botnet* de medidores. Em ambos os casos, é necessário encontrar e explorar vulnerabilidades não corrigidas do *firmware* do medidor. Como explicado anteriormente, o uso de assinaturas digitais para autenticação de *firmware* pode não ser suficiente para proteger um medidor inteligente [McDaniel and McLaughlin 2009]. A partir de experiências passadas, sabemos que falhas de segurança são inevitáveis, especialmente quando se lida com um sistema em que a pirataria pode ser tão facilmente monetizada. Assim, a segurança da rede de comunicação deve ser realizada com base em técnicas de segurança diferentes.

4.3.3.4. Ataques contra a Privacidade do Usuário em Sistemas de Distribuição

Tendo descrito possíveis ataques à disponibilidade de serviço e ataques contra a integridade do serviço que poderia ocorrer em sistemas de distribuição, agora serão descritos os problemas de segurança contra a privacidade do usuário. Na rede elétrica tradicional, funcionários da operadora coletam mensalmente as informações do medidor. Com os avanços promovidos pela AMI, os dados de medição tornaram-se mais detalhados e coletados em intervalos de tempo mais curtos [Siddiqui et al. 2012]. Embora os dados detalhados e granulares sejam importantes para permitir vários serviços de *smart grid*, eles criam grandes vulnerabilidades de privacidade. Os principais tipos de ataques contra a privacidade do usuário são a espionagem e a análise de tráfego. Ambos tiram proveito da comunicação sem fio da NAN para obter detalhes pessoais da vida do usuário. Espionagem é a escuta não autorizada de uma conversa privada, neste caso a interceptação de um canal de comunicação sem fio. Na análise de tráfego, o padrão de tráfego é monitorado com a intenção de inferir hábitos diários dos usuários.

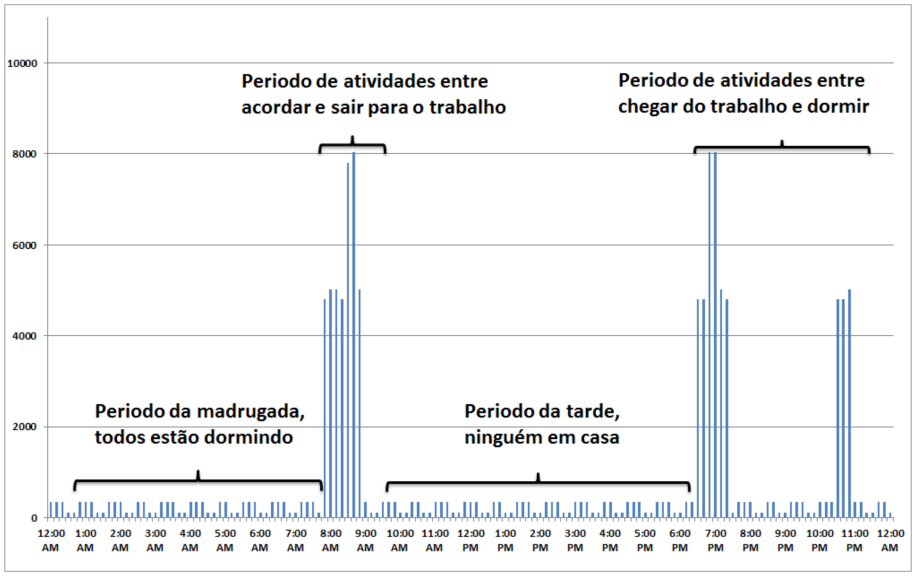


Figura 4.11. Exemplo de dados de medição do consumo de energia que poderiam ser usados para espionar a rotina do usuário [Molina-Markham et al. 2010].

A troca periódica de dados entre aparelhos inteligentes e o EMS (*Energy Management System*) permite ao usuário monitorar seu consumo em tempo real e controlar remotamente dispositivos domésticos. No entanto um adversário poderia escutar esta comunicação, adquirir o consumo do dispositivo e usar um algoritmo de criação de perfil de carga para identificar qual o dispositivo ligado. Cada dispositivo tem uma assinatura de carga, um comportamento elétrico único, que pode ser utilizado num processo de reconhecimento de dispositivos elétricos [Rahimi et al. 2011], e numa análise mais aprofundada, o intruso pode obter ainda conteúdos multimídia, como o canal de televisão a que o usuário está assistindo [Greveler et al. 2012]. A questão da privacidade é séria, não só por causa da exposição da vida pessoal do usuário, mas também pelo fato de que esta informação pode comprometer a própria segurança do usuário. Através de análise de tráfego, um perfil de consumo pode ser traçado e detalhes de rotina de um usuário podem

ser inferidos, a que horas ele acorda, que horas vai dormir, se está viajando, quais pessoas estão em casa em um hora específica, como mostrado na Figura 4.11. Esta informação pode ser usada para furto, roubo e até mesmo sequestro.

A detecção destes tipos de ataques torna-se difícil devido à sua natureza passiva. O adversário deseja roubar informações sem alteração de dados e sem modificar o funcionamento do sistema. Assim, a prevenção é mais importante do que a detecção.

4.4. Soluções e Recomendações

Depois de descrever os principais ataques à comunicação das redes elétricas inteligentes, as principais soluções para criar uma arquitetura de comunicação integrada, que forneça confiabilidade, privacidade e não repúdio serão abordadas.

4.4.1. Subestações e Criptografia na AMI

Como discutido anteriormente, a comunicação das redes elétricas inteligentes é vulnerável a diversos tipos de ataques. A comunicação de dados sem criptografia é uma das razões que aumentam a vulnerabilidade. A fim de entender melhor esse cenário e onde a criptografia pode ser usada, é necessário compreender que tipo de protocolos para subestações e *smart grid* são utilizados. Nesta seção, iremos classificá-los em três tipos. O primeiro tipo é chamado protocolo SCADA, que como já citado anteriormente é responsável pela comunicação entre os sistemas supervisórios e dispositivos como medidores inteligentes e IEDs. Os protocolo SCADA geralmente têm restrições de tempo mais brandas do que os outros já que incluem a interação do usuário. Portanto, a interface do usuário é parte do sistema supervisório e os atrasos dependem da atividade do usuário. Assim, atrasos de 200 ms a 1000 ms são bem aceitos na rede.

O segundo tipo de protocolo é chamado de protocolo de proteção. Ao contrário do tipo anterior, este tipo de protocolo possui severas limitações de tempo que variam de 3 ms a 100 ms de acordo com os requisitos do cenário. Neste caso, a fim de evitar falhas ou para limitar a interrupção de um serviço, devido a uma falha elétrica, as mensagens de proteção de rede são enviadas depois de um evento elétrico. Falhas podem afetar uma grande parte do sistema elétrico muito rapidamente, assim todas as mensagens de proteção exigem rígidas restrições de tempo. É importante destacar que os dispositivos de proteção (por exemplo, IEDs) podem detectar condições de falha e enviar mensagens de proteção para outro dispositivo (por exemplo, um disjuntor inteligente ou um IED que pode operar um disjuntor) apenas nos sistemas modernos e mais recentes. Nos sistemas tradicionais, um dispositivo de proteção detecta condições de falha e opera diretamente um disjuntor através de cabos de controle, sem uma rede de comunicação.

O terceiro tipo é destinado a medidas e amostragem de valores. A ideia inclui a amostragem de valores instantâneos dos sistemas de potência, principalmente correntes primárias e tensões, e a transmissão através da rede. Estes valores são publicados na rede e dispositivos de proteção ou controle (ou qualquer dispositivo que possa fazer uso deles) são capazes de assiná-los. Valores amostrados são usados pelos dispositivos de proteção e controle para a identificação de falhas elétricas. Portanto, as restrições de tempo são tão rígidas quanto no protocolo de proteção, ou mais, conforme for necessário para a identificação da falha.

As limitações de tempo para envio de mensagens são um dos principais entraves para o uso de criptografia nesse cenário. Em geral, a criptografia ou encriptação é a transformação da informação a partir de um estado compreensivo para um estado aparente absurdo. Assim, a criptografia transmite dados de uma forma particular ao destinatário, de maneira que apenas ele possa processá-los. O remetente de uma mensagem encriptada compartilha, somente com o receptor pretendido, a técnica de decodificação necessária para recuperar a informação original. Medidas criptográficas incluem a criptografia de chave simétrica e criptografia de chave assimétrica. Os métodos mais antigos necessitam de mais recursos computacionais, como AES (*Advanced Encryption Standard*) e DES (*Data Encryption Standard*) [Mishra et al. 2016]. A gestão de chave de segurança é essencial para a criptografia de informações.

No entanto, a criptografia acrescenta atraso na rede de comunicação. Como a codificação e a descodificação são realizadas nos pacotes, o tempo total a partir de um evento e uma operação, por exemplo, pode aumentar consideravelmente. Reconhecendo este fato, a maioria das soluções de criptografia não são adequadas para utilização em protocolos com limitações de tempo rígidas como os protocolos de medidas e amostragem de valores e protocolos de proteção. É necessário que os métodos de criptografia não só satisfaçam requisitos de desempenho em tempo real desses protocolos, mas também garantam a segurança da mensagem. Como existem poucos trabalhos sobre o assunto publicados na literatura [Fangfang et al. 2013], é uma boa oportunidade de pesquisa: um método para proporcionar segurança combinada com a qualidade do serviço. Testes exaustivos devem ser feitos para garantir que as soluções com criptografia não excedam requisitos de atraso em *smart grids*. Em [Fangfang et al. 2013], os autores mostram por simulação com o software OPNET que com um método de criptografia híbrido, é possível. No entanto, testes mais detalhados precisam ser feitos.

Por outro lado, a criptografia pode ser adicionada aos protocolos SCADA, sem problemas, uma vez que este tipo de protocolo não tem muitas limitações de atraso. Vários pesquisadores propuseram métodos de criptografia e esquemas de gerenciamento de chaves para SCADA. Isso ocorre porque os protocolos SCADA não foram projetados considerando a necessidade de segurança. Como mostrado em [Amoah et al. 2016], muitos pesquisadores têm proposto soluções para este fim, mas ainda há muito trabalho a ser feito. Soluções propostas na literatura são muito específicas, com muitas peculiaridades. Algumas soluções são apresentadas a seguir, juntamente com soluções para comunicação segura.

4.4.2. Solução de Padronização para a Segurança em Subestações

Devido a razões históricas, as questões de segurança cibernética não fazem parte dos protocolos industriais. DNP3, 60870-5 e IEC 61850 foram publicados quando a segurança não era uma grande preocupação industrial. Para superar este problema, o IEC padrão 62351 foi desenvolvido pelo Comitê de IEC Técnica (TC) 57, a fim de fornecer os requisitos de segurança em redes de automação de energia. Na verdade, a IEC 62351 já utiliza os métodos atuais, tanto quanto possível. Por exemplo, utiliza o protocolo TLS (*Transport Layer Security*), a fim de preservar a integridade das mensagens de acordo com um esquema forte de gestão de identidade. Além disso, ele propõe o uso de RBAC (*Role-Based Access Control*). Isto significa que não só pretende identificar de forma se-

gura todas as entidades do sistema, mas também definir políticas de controle de acesso baseadas na função da entidade no sistema. Os objetivos do padrão também incluem integridade, confidencialidade, prevenção de falsificação, detecção de intrusão, autenticação por meio de certificados digitais, e assim por diante. Este padrão é dividido em 10 partes, como ilustrado na Figura 4.12.

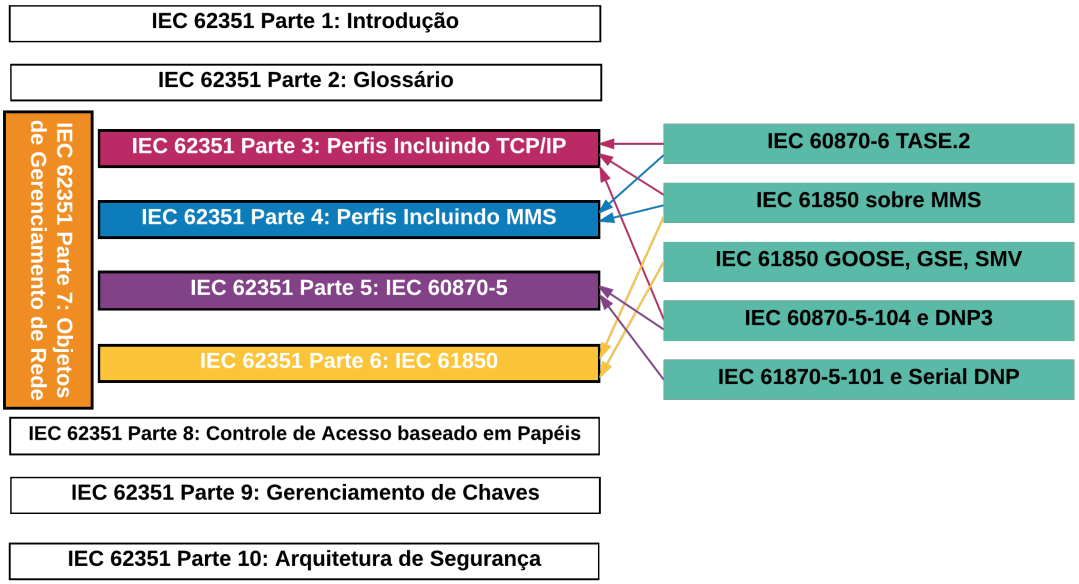


Figura 4.12. Estrutura da norma IEC 62351.

Uma solução bem conhecida já aplicada é um Sistema de Detecção de Intrusão (*Intrusion Detection System* - IDS). A finalidade de um IDS é detectar comportamento anormal na rede digitalizando pacotes e gerando alertas para os operadores. IDS são normalmente classificados como *network-based* ou *host-based*. Um IDS baseado em rede monitora o tráfego de rede local e tem acesso a todos os pacotes transmitidos, enquanto um IDS baseado em host analisa os pacotes em um ou mais servidores individualmente [Sun et al. 2016]. Um trabalho recente de Mishra et al. [Mishra et al. 2016] demonstrou um método ideal para a digitalização de pacotes como uma defesa contra ataques a preços acessíveis, oferecendo segurança, sem comprometer os requisitos rigorosos de QoS.

4.4.3. Soluções de Segurança para AMI

Soluções de segurança para AMI estão em discussão e há um elevado número de propostas sobre a forma de garantir a segurança na rede. Nesta seção, vamos mostrar soluções e sistemas de proteção para os ataques contra estruturas cibernéticas internas de uma rede de distribuição e os desafios de melhorar a segurança neste cenário.

O ataque de bloqueio de sinal foi apresentado como uma ameaça à disponibilidade da rede elétrica. Alguns estudos propõem soluções para este tipo de ataque à rede sem fio, a maioria deles são técnicas baseadas em salto de frequência. Por exemplo, há uma proposta que sugere o uso de múltiplos canais de frequências alternativas [Aravinthan et al. 2011]. Se os medidores detectarem interferências no canal atual, to-

dos os medidores se movem através de uma sequência aleatória pré-definida e comum de canais. Quando um medidor completa o processo de autenticação, recebe a sequência de saltos de frequência através de um canal encriptado. Outra proposta consiste em um esquema aleatório de espalhamento espectral designado por FQR (*Frequência Quorum Rendezvous*) [Lee et al. 2011]. FQR explora um sistema de quorum que permite que cada nó construa uma sequência de saltos de forma independente durante a fase de estabelecimento de chave. A propriedade de intersecção do sistema de quorum garante que um par de nós se encontre dentro de um período limitado de tempo, durante o qual eles compartilham uma chave comum usada para comunicações futuras de espalhamento espectral. Este mecanismo não só evita interferência, mas também espionagem através de escuta do canal sem fio.

A prevenção de fraudes e injeção de dados falsos pode ser alcançada com forte criptografia fim a fim em canais de comunicação da rede HAN (*Home Area Network*) e AMI. Como as restrições de atraso para esta aplicação não são críticas, muitas opções de criptografia fim a fim utilizadas na Internet estão facilmente disponíveis. Uma lista de trabalhos relacionados com a prevenção de ataque com dados falsos podem ser encontrados em [Sharma and Saini 2015]. Autenticação também é de grande importância, uma vez que dados falsos ou medidores fraudulentos podem ser inseridos na rede dando ao invasor a capacidade de executar comandos e degradar serviços. Nicanfar et al. [Nicanfar et al. 2014] apresentam uma rede inteligente de autenticação mútua (SGMA) que fornece uma autenticação eficiente entre medidor inteligente e servidor de autenticação usando senhas, e um protocolo de rede inteligente de gerenciamento de chaves (SGKM) utilizando a infraestrutura de chave pública (PKI). Também foram propostas novas melhorias para proteger a privacidade do usuário. Neste caso, um mecanismo obscurece parcialmente o perfil de carga do usuário usando uma bateria recarregável, e protege a privacidade do usuário [Varodayan and Khisti 2011]. O consumo relatado pelo medidor inteligente para a empresa concessionária é uma combinação de aparelhos e o consumo da bateria. A qualquer momento, a bateria pode realizar uma combinação das seguintes ações (ou nenhuma delas) sujeitas à sua capacidade: transmitir energia diretamente do utilitário para os aparelhos; armazenar energia da concessionária para uso futuro; entregar a energia armazenada anteriormente para os aparelhos. Desta forma, o carregamento e a descarregamento de uma bateria pode manipular a carga de saída, obscurecendo a informação do consumidor real. Outras soluções para a privacidade são baseadas em esquemas de gerenciamento de chaves para proteger o conteúdo que está sendo transmitido no interior da casa entre o medidor e o sistema de controle [Kazienko et al. 2015].

4.5. Direções para Futuras Pesquisas

Neste capítulo, vulnerabilidades tanto de cliente e medidores inteligentes quanto de subestações e centros de controle foram abordadas. Os ataques e ameaças e as suas consequências foram descritos. Além disso, as possíveis soluções encontradas na literatura foram destacadas, o que pode nos ajudar a alcançar uma rede inteligente segura. No entanto, a implantação de uma rede segura de comunicação para *smart grid* ainda é um enorme desafio. Algumas linhas de pesquisa incluem:

- Propostas como IEC 62351 prometem resultados, mas elas exigem testes exausti-

vos e experimentações. Elas devem ser testadas com protocolos industriais para atestar que a qualidade dos serviços exigida por aplicações de energia não seja afetada. Além disso, a proposta padrão está sendo atualizada com novos métodos ou evolução dos métodos já conhecidos, o que confirma a necessidade de avaliação constante.

- A padronização é também uma preocupação principal. A arquitetura de rede inteligente é complexa e composta por diferentes tipos de domínios e redes. O sucesso da implantação comercial das redes elétricas inteligentes depende de mecanismos padrão que permitam que diferentes fornecedores possam interoperar em um formato uniforme. Um padrão universal pode ser utilizado a fim de proporcionar interoperabilidade, a flexibilidade, a redução de custos, e assim por diante. A utilização do mesmo modelo de informação irá melhorar a comunicação e, principalmente, pesquisas de segurança. Há indícios de uso do IEC 61850, no entanto ainda existem muitas questões a serem avaliadas.

- O estabelecimento de uma estrutura universal para comunicação segura das redes inteligentes é muito importante. Esta estrutura deve levar em consideração a proteção "defesa em profundidade"[Lüders 2011]. Assim, cada camada de comunicação deve ser tratada, bem como o hardware e software. Técnicas de segurança de TI tradicionais também devem ser abordadas no contexto de redes inteligentes, incluindo questões como:

  - Um processo fácil e barato para aplicar *patches* de segurança em IEDs e medidores inteligentes;
  - A utilização de estruturas de gerenciamento de identidade para fornecer autenticação e autorização segura;
  - O uso de sistemas de gerenciamento de configuração para IEDs e medidores inteligentes;
  - A utilização de novos *firewalls* e técnicas de detecção de intrusão em subestações e redes de AMI.
  - O uso de técnicas de *big data* para descobrir informações importantes entre os dados coletados de medidores inteligentes para ajudar a descobrir *botnets*, roubo de energia, e até mesmo ações de terrorismo.

- A proposta de métodos de autenticação e integridade mais seguras para a comunicação *multicast*, sem incorrer nos requisitos de alto poder de processamento ou elevados atrasos de comunicação. Isto é de especial importância para proporcionar uma comunicação entre IEDs que seja mais resistente contra os invasores externos e internos.

- A proposta de uma solução completa, que compreende a interoperabilidade entre diferentes sistemas de criptografia, também é necessária. Diferentes tecnologias de comunicação e protocolos são usados na infraestrutura de rede inteligente que resulta em requisitos de criptografia exclusivos para cada um. Uma abordagem segura e interoperável é essencial.

- O equilíbrio entre a disponibilidade da informação e preservação da privacidade não é trivial e é uma direção de pesquisa muito interessante. Enquanto um grande conjunto de informações resulta em decisões mais inteligentes e melhores otimizações, também representa uma ameaça à privacidade do usuário. Informações de medidores inteligentes tornam possível inferir o comportamento do consumidor, o que possivelmente ofenderá os consumidores.

4.6. Conclusão

O avanço da comunicação para *smart grid* estabelece novos desafios de segurança. Como o cenário em subestações e na mudança dos sistemas de distribuição, velhos conceitos caíram por terra. Atualmente, atacantes são uma realidade para redes de campo em subestações, mesmo com o uso de firewalls, redes privadas virtuais (VPN), sistemas de detecção de intrusão (IDS), e algumas técnicas de criptografia.

Não há solução definitiva para a segurança nas redes de comunicação de *smart grid*, bem como esta solução completa não existe para a Internet também. A segurança deve sempre evoluir, porque os *hackers* estão sempre à procura de novas brechas. Processos legados para gerenciar redes ICS já não são aceitáveis. Práticas como a não utilização de *firewalls*, não usar anti-vírus, não aplicação de *patches*, usar configurações padrão e contas padrão, a não utilização de certificação digital, etc. já não são aceitáveis. Além disso, os fornecedores devem assumir a responsabilidade para o desenvolvimento de dispositivos seguros e robustos. Muito esforço tem sido colocado nos últimos anos para o desenvolvimento de dispositivos que sejam seguros contra explosões ou falhas no sistema elétrico e que sejam robustos a diferentes cenários de falhas de energia. No entanto, esses dispositivos não são testados em casos de abuso e não podem ser considerados seguros.

Muitas dessas observações partem do fato de que existem muitos especialistas em segurança de rede de TI, mas apenas alguns que entender tanto de segurança de rede e sistemas de controle industrial. Assim, existe uma forte necessidade de formar mais profissionais capacitados para trabalhar nesta área multidisciplinar.

Referências

[1646 2004] 1646, I. (2004). Ieee standard communication delivery time performance requirements for electric power substation automation.

[Amoah et al. 2016] Amoah, R., Camtepe, S., and Foo, E. (2016). Securing dnp3 broadcast communications in scada systems. *IEEE Transactions on Industrial Informatics*, 12(4):1474–1485.

[Aravinthan et al. 2011] Aravinthan, V., Namboodiri, V., Sunku, S., and Jewell, W. (2011). Wireless ami application and security for controlled home area networks. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–8. IEEE.

[Assante 2016] Assante, M. (2016). Confirmation of a coordinated attack on the ukrainian power grid. *Online]: https://ics. sans. org/blog/2016/01/09/confirmation-of-a-coordinated-attack-on-the-ukrainianpower-grid*.

[Bayat et al. 2015] Bayat, M., Arkian, H. R., and Aref, M. R. (2015). A revocable attribute based data sharing scheme resilient to dos attacks in smart grid. *Wireless Networks*, 21(3):871–881.

[Bayod-Rújula 2009] Bayod-Rújula, A. A. (2009). Future development of the electricity systems with distributed generation. *Energy*, 34(3):377 – 383. {WESC} 2006 6th World Energy System Conference Advances in Energy Studies 5th workshop on Advances, Innovation and Visions in Energy and Energy-related Environmental and Socio-Economic Issues.

[Budka et al. 2010] Budka, K., Deshpande, J., Hobby, J., Kim, Y.-J., Kolesnikov, V., Lee, W., Reddington, T., Thottan, M., White, C., Choi, J.-I., Hong, J., Kim, J., Ko, W., Nam, Y.-W., and Sohn, S.-Y. (2010). GERI - Bell Labs smart grid research focus: Economic modeling, networking, and security & privacy. In *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 208–213.

[Budka et al. 2014] Budka, K. C., Deshpande, J. G., Thottan, M., et al. (2014). Communication networks for smart grids. In *Computer Communications and Networks*. Springer.

[Cheung et al. 2007] Cheung, H., Hamlyn, A., Wang, L., Yang, C., and Cheung, R. (2007). Computer network security strategy for coordinated distribution system operations. In *Power Engineering, 2007 Large Engineering Systems Conference on*, pages 279–283.

[Chim et al. 2011] Chim, T., Yiu, S., Hui, L., and Li, V. (2011). PASS: Privacy-preserving authentication scheme for smart grid network. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 196 –201.

[Cleveland 2008] Cleveland, F. (2008). Cyber security issues for advanced metering infrasttructure (AMI). In *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1 – 5.

[DoE 2010] DoE (2010). Communication requirements of smart grid. U.S. Department of Energy (DoE).

[East et al. 2009] East, S., Butts, J., Papa, M., and Shenoi, S. (2009). A taxonomy of attacks on the dnp3 protocol. In *International Conference on Critical Infrastructure Protection*, pages 67–81. Springer.

[EPRI 2009] EPRI (2009). Report to nist on the smart grid interoperability standards roadmap. Electric Power Research Institute.

[Falliere et al. 2011] Falliere, N., Murchu, L. O., and Chien, E. (2011). W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5:6.

[Fangfang et al. 2013] Fangfang, W., Huazhong, W., Dongqing, C., and Yong, P. (2013). Substation communication security research based on hybrid encryption of des and rsa. In *Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on*, pages 437–441. IEEE.

[Finster and Baumgart 2015] Finster, S. and Baumgart, I. (2015). Privacy-aware smart metering: A survey. *IEEE Communications Surveys & Tutorials*, 17(2):1088–1101.

[Giani et al. 2011] Giani, A., Bitar, E., Garcia, M., McQueen, M., Khargonekar, P., and Poolla, K. (2011). Smart grid data integrity attacks: Characterizations and countermeasures. *Cyber and Physical Security and Privacy*, pages 232–237.

[Greveler et al. 2012] Greveler, U., Glösekötterz, P., Justusy, B., and Loehr, D. (2012). Multimedia content identification through smart meter power usage profiles. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[Group 2010] Group, C. S. W. (2010). The smart grid interoperability panel - guidelines for smart grid cyber security. NISTIR 7628, pp. 1-597.

[Gungor et al. 2011] Gungor, V., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., and Hancke, G. (2011). Smart grid technologies: Communication technologies and standards. *IEEE Transactions on Industrial Informatics*, 7(4):529–539.

[IEC 2007] IEC (1988- 2007). IEC 60870-5: Telecontrol equipment and systems - Part 5: Transmission protocols. Technical report, International Electrotechnical Commission.

[IEC 2013] IEC (2002- 2013). IEC 61850: Communication networks and systems for power utility automation. Technical Report IEC 61850, International Electrotechnical Commission.

[IEC 2009] IEC, T. (2009). 57. communication networks and systems in substations–part 7–420: basic communication structure–distributed energy resources logical nodes. *Int. Electrotech. Comm.*

[IEEE 2012] IEEE (2012). Ieee standard for electric power systems communications-distributed network protocol (dnp3). pages 1–821. IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010).

[Kazienko et al. 2015] Kazienko, J. F., Moraes, I. M., Albuquerque, C. V., et al. (2015). On the performance of a secure storage mechanism for key distribution architectures in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015:1.

[Khaitan et al. 2015] Khaitan, S. K., McCalley, J. D., and Liu, C. C. (2015). *Cyber Physical Systems Approach to Smart Electric Power Grid*. Springer.

[Kounev et al. 2016] Kounev, V., Lévesque, M., Tipper, D., and Gomes, T. (2016). Reliable communication networks for smart grid transmission systems. *Journal of Network and Systems Management*, pages 1–24.

[Kush et al. 2014] Kush, N., Ahmed, E., Branagan, M., and Foo, E. (2014). Poisoned goose: exploiting the goose protocol. In *Proceedings of the Twelfth Australasian Information Security Conference-Volume 149*, pages 17–22. Australian Computer Society, Inc.

[Lee et al. 2011] Lee, E.-K., Oh, S. Y., and Gerla, M. (2011). Frequency quorum rendezvous for fast and resilient key establishment under jamming attack. *ACM SIGMOBILE Mobile Computing and Communications Review*, 14(4):1–3.

[Li and Han 2011] Li, H. and Han, Z. (2011). Manipulating the electricity power market via jamming the price signaling in smart grid. In *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, pages 1168–1172. IEEE.

[Li et al. 2015] Li, Q., Ross, C., Yang, J., Di, J., Balda, J. C., and Mantooth, H. A. (2015). The effects of flooding attacks on time-critical communications in the smart grid. In *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society*, pages 1–5. IEEE.

[Lopes et al. 2015a] Lopes, Y., Fernandes, N. C., and Muchaluat-Saade, D. C. (2015a). Geração Distribuída de Energia: Desafios e Perspectivas em Redes de Comunicação. In *Minicursos do XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 55–109. Sociedade Brasileira de Computação (SBC), "Vitória, Espírito Santo, Brasil", 1 edition.

[Lopes et al. 2012] Lopes, Y., Frazão, R. H., Molano, D. A., dos Santos, M. A., Calhau, F. G. a., Bastos, C. A. M., Martins, J. S. B., and Fernandes, N. C. (2012). Smart Grid e IEC 61850: Novos Desafios em Redes e Telecomunicações para o Sistema Elétrico. In *Minicursos do XXX Simpósio Brasileiro de Telecomunicações*, pages 1–44. 1 edition.

[Lopes et al. 2015b] Lopes, Y., Muchaluat-Saade, D. C., Fernandes, N. C., and Fortes, M. Z. (2015b). Geese: A traffic generator for performance and security evaluation of iec 61850 networks. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pages 687–692. IEEE.

[Lüders 2011] Lüders, S. (2011). Why control system cybersecurity sucks. GovCERT.NL Symposium.

[McDaniel and McLaughlin 2009] McDaniel, P. and McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *IEEE Security and Privacy*, 7(3):75–77.

[Mishra et al. 2016] Mishra, S., Dinh, T. N., Thai, M. T., Seo, J., and Shin, I. (2016). Optimal packet scan against malicious attacks in smart grids. *Theoretical Computer Science*, 609:606–619.

[Molina-Markham et al. 2010] Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., and Irwin, D. (2010). Private memoirs of a smart meter. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, pages 61–66. ACM.

[Neuman and Tan 2011] Neuman, C. and Tan, K. (2011). Mediating cyber and physical threat propagation in secure smart grid architectures. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 238–243.

[Nicanfar et al. 2014] Nicanfar, H., Jokar, P., Beznosov, K., and Leung, V. C. (2014). Efficient authentication and key management mechanisms for smart grid communications. *IEEE systems journal*, 8(2):629–640.

[NIST 2010] NIST (2010). Nist 7628 - guidelines for smart grid cyber security vol. 1: smart grid cyber security strategy, architecture, and high-level requirements. National Institute of Standards and Technology.

[Noce et al. 2016] Noce, J., Lopes, Y., Muchaluat-Saade, D. C., Fernandes, N. C., and Albuquerque, C. (2016). Identificando falhas de segurança na rede de comunicação de subestações digitalizadas em redes elétricas inteligentes utilizando GEESE 2.0. In *XXI Congresso Brasileiro de Automática (CBA)*, pages 1–6. SBA.

[Organization 2005] Organization, M. (2005). *Modbus protocol*. www.modbus.org/specs.php.

[Pan et al. 2014] Pan, J., JAIN, R., and Paul, S. (2014). A survey of energy efficiency in buildings and microgrids using networking technologies. *IEEE Communications Surveys Tutorials*, (3):1709–1731.

[Patel et al. 2011] Patel, A., Aparicio, J., Tas, N., Loiacono, M., and Rosca, J. (2011). Assessing communications technology options for smart grid applications. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 126–131.

[PUB 2006] PUB, F. (2006). Minimum security requirements for federal information and information systems.

[Rahimi et al. 2011] Rahimi, S., Chan, A. D., and Goubran, R. A. (2011). Usage monitoring of electrical devices in a smart home. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5307–5310. IEEE.

[Rahman et al. 2012] Rahman, M., Bera, P., and Al-Shaer, E. (2012). SmartAnalyzer: A noninvasive security threat analyzer for AMI smart grid. In *Proceedings IEEE INFOCOM*, pages 2255 – 2263.

[Rodofile et al. 2015] Rodofile, N., Radke, K., and Foo, E. (2015). Real-time and interactive attacks on dnp3 critical infrastructure using scapy.

[Sharma and Saini 2015] Sharma, K. and Saini, L. M. (2015). Performance analysis of smart metering for smart grid: An overview. *Renewable and Sustainable Energy Reviews*, 49:720–735.

[Siddiqui et al. 2012] Siddiqui, F., Zeadally, S., Alcaraz, C., and Galvao, S. (2012). Smart grid privacy: Issues and solutions. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–5. IEEE.

[Sun et al. 2016] Sun, C.-C., Liu, C.-C., and Xie, J. (2016). Cyber-physical system security of a power grid: State-of-the-art. *Electronics*, 5(3):40.

[Ur-Rehman et al. 2015] Ur-Rehman, O., Zivic, N., and Ruland, C. (2015). Security issues in smart metering systems. In *Smart Energy Grid Engineering (SEGE), 2015 IEEE International Conference on*, pages 1–7. IEEE.

[Varodayan and Khisti 2011] Varodayan, D. and Khisti, A. (2011). Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1932–1935. IEEE.

[Wang et al. 2011] Wang, J., Yang, X., and Long, K. (2011). Web DDoS detection schemes based on measuring user's access behavior with large deviation. In *IEEE Global Telecommunications Conference (GLOBECOM 2011)*, pages 1 – 5.

[Wei and Wang 2014] Wei, M. and Wang, W. (2014). Greenbench: A benchmark for observing power grid vulnerability under data-centric threats. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2625–2633.

[Wei and Wang 2016] Wei, M. and Wang, W. (2016). Data-centric threats and their impacts to real-time communications in smart grid. *Computer Networks*, 104:174–188.

[Wilhoit 2013] Wilhoit, K. (2013). The scada that didn't cry wolf. *Trend Micro Inc., White Paper*.

[Yan et al. 2011] Yan, Y., Qian, Y., and Sharif, H. (2011). A secure and reliable in-network collaborative communication scheme for advanced metering infrastructure in smart grid. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 909 –914.

[Yoo and Shon 2015] Yoo, H. and Shon, T. (2015). Novel approach for detecting network anomalies for substation automation based on iec 61850. *Multimedia Tools and Applications*, 74(1):303–318.

[Zhu et al. 2011] Zhu, T., Xiao, S., Ping, Y., Towsley, D., and Gong, W. (2011). A secure energy routing mechanism for sharing renewable energy in smart microgrid. In *2011 IEEE International Conference on Smart Grid Communications (IEEE SmartGridComm)*, pages 143–148.

SBSeg16

XVI SIMPÓSIO BRASILEIRO
EM SEGURANÇA DA INFORMAÇÃO
E DE SISTEMAS COMPUTACIONAIS

7 A 10 DE NOVEMBRO | NITERÓI | RJ

sbseg2016.ic.uff.br

Realizado por:

SBC Sociedade Brasileira de Computação

Instituto de Computação

Universidade Federal Fluminense

Apoiado por:

COPPE UFRJ

UFRJ

CESEG

Patrocinado por:

nic.br Núcleo de Informação e Coordenação do Ponto BR

cgi.br Comitê Gestor da Internet no Brasil

intel

FAPERJ

CNPq

CAPES

proppi

CLAVIS segurança da informação

Google