COPPE
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
UFRJ

MIGRAÇÃO AO VIVO SEM PERDA DE PACOTES E PROCESSAMENTO HOMOMÓRFICO DE NÚMEROS INTEIROS PARA PLATAFORMAS DE TESTES COMPARTILHADAS

Pedro Silveira Pisa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Otto Carlos Muniz Bandeira Duarte
Michel Abdalla

Rio de Janeiro
Agosto de 2013

MIGRAÇÃO AO VIVO SEM PERDA DE PACOTES E PROCESSAMENTO HOMOMÓRFICO DE NÚMEROS INTEIROS PARA PLATAFORMAS DE TESTES COMPARTILHADAS

Pedro Silveira Pisa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr. Ing.

Dr. Michel Abdalla, Ph.D.

Prof. Pedro Braconnot Velloso, Ph.D.

Prof. Ricardo Dahab, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2013

Pisa, Pedro Silveira

Migração ao Vivo sem Perda de Pacotes e Processamento Homomórfico de Números Inteiros para Plataformas de Testes Compartilhadas/Pedro Silveira Pisa. – Rio de Janeiro: UFRJ/COPPE, 2013.

XX, 98 p.: il.; 29, 7cm.

Orientadores: Otto Carlos Muniz Bandeira Duarte
Michel Abdalla

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2013.

Referências Bibliográficas: p. 89 – 98.

1. Internet do Futuro. 2. Gerência de Redes Virtualizadas. 3. Migração de Roteadores Virtuais. 4. Privacidade. 5. Segurança. 6. Criptografia Homomórfica. I. Duarte, Otto Carlos Muniz Bandeira *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha família e à minha namorada Claudia

Agradecimentos

Aos meus pais, Wilson e Lucia, por todo amor, incentivo e orientação passados em todos os anos de minha vida. À minha irmã Heloisa, pelo companheirismo e amizade. À minha avó, Mercedes, pelo amor, pelos cuidados e pela dedicação. À minha namorada, Claudia, pelo amor, pelos momentos felizes e por toda a compreensão nos momentos de dificuldade. Sem eles, a conclusão dessa importante fase nunca aconteceria.

Aos amigos do Mestrado, em especial para o Hugo Eiji, Diogo Ferrazani, Lyno Ferraz, Pedro Coutinho e Daniel Dias, pelos momentos de diversão, como as risadas e brincadeiras dentro e fora das salas de aula e dos laboratórios, e de concentração, como nos estudos para a prova e reuniões de trabalhos e por toda a ajuda que me deram durante o mestrado.

Ao professor e orientador Otto, responsável por grande parte da minha formação acadêmica e profissional, e ao co-orientador Michel Abdala, por seus conselhos e orientação. Aos professores Luis Henrique, Miguel, Igor, Pedro e Natalia pelos ensinamentos e pelas dicas. Aos professores dos programas de Sistemas e Elétrica da COPPE, que ministraram aulas me ensinando mais do que o conhecimento da Engenharia e da pesquisa científica; ensinaram lições de vida e me deram diferentes de visões de mundo, enriquecendo minha formação pessoal e profissional.

Aos professores e pesquisadores Otto Carlos Muniz Bandeira Duarte, Michel Abdalla, Pedro Braconnot Velloso e o Ricardo Dahab pela presença na banca examinadora e pela contribuição neste trabalho.

Aos amigos e colegas durante toda a minha passagem pelo GTA, Lyno, Hugo, Diogo, Natalia, Marcelo, Filipe Barretto, Igor, Miguel, Carlo, Daniel Dias, Rafael Santos, Rodrigo Couto, Danilo, Carina e Reinaldo pelos conselhos, pela troca de experiências e, para alguns, pela grande ajuda neste trabalho. A todos os alunos de iniciação científica, que ajudaram no desenvolvimento deste trabalho com ideias desprendidas de preconceitos e padrões pré-estabelecidos. Agradeço ainda aos alunos atuais do GTA, que certamente farão o grupo ser a referência em pesquisa científica que sempre foi, mesmo eu não tendo tido a oportunidade de conhecê-los melhor.

Aos amigos e colegas da graduação, Hugo Eiji, Daniel Vega, Diogo Menezes, Pedro Coutinho, João Pedro Francese, Leonardo Arnt, Gustavo Fernandes, Thiago

Xavier, Renan Bernardo, João Luiz Ferreira, Victor Bursztyn e Marden Braga por todo o apoio e conhecimento trocados, mesmo que não nos encontremos todos mais com tanta frequência.

De forma geral, agradeço a todos que me incentivaram, contribuindo de forma direta ou indireta para a minha formação pessoal e profissional e pelo aprendizado absorvido nesses dois anos de mestrado.

Agradeço às instituições CNPQ, FINEP, PIBIC, CAPES, FUNTTEL e FAPERJ e à empresa UOL pelo financiamento da pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MIGRAÇÃO AO VIVO SEM PERDA DE PACOTES E PROCESSAMENTO HOMOMÓRFICO DE NÚMEROS INTEIROS PARA PLATAFORMAS DE TESTES COMPARTILHADAS

Pedro Silveira Pisa

Agosto/2013

Orientadores: Otto Carlos Muniz Bandeira Duarte
Michel Abdalla

Programa: Engenharia Elétrica

Esta dissertação aborda duas questões fundamentais para a Internet do Futuro e as plataformas de testes compartilhadas: a gerência de redes virtualizadas e o processamento de dados encriptados através da criptografia homomórfica.

A gerência de redes virtualizadas adiciona complexidade à gerência de redes tradicionais, sobretudo devido ao grande aumento das variáveis a serem monitoradas e das diferentes possiblidades de alocação de recursos. A primeira parte dessa dissertação apresenta uma ferramenta de gerência de plataformas de testes chamada VNEXT (*Virtual NEtwork Management for Xen-based Testbeds*), criada pelo Grupo de Teleinformática e Automação (GTA). São apresentados um mecanismo de migração ao vivo sem perda de pacotes, uma reformulação arquitetural do controlador da ferramenta e uma interface de visualização das topologias de rede.

A encriptação homomórfica, tema da segunda parte da dissertação, é um método de encriptação para processar dados cifrados sem a necessidade de descriptografá-los. Essa técnica agrega privacidade e segurança ao processamento em ambientes não confiáveis, como as plataformas de testes compartilhadas e a computação em nuvem. Um esquema de encriptação homomórfica para números inteiros de tamanho arbitrário é proposto nessa dissertação. A proposta é uma extensão do esquema de encriptação proposto por Dijk, Gentry, Halevi e Vaikuntanathan (DGHV) [1] e ambos os esquemas foram implementados. Os resultados demonstram que a proposta aumenta a eficiência na encriptação e permite a operação de números grandes, mantendo a quantidade total de dados armazenados. Observa-se ainda que a geração das chaves públicas é custosa.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

LIVE MIGRATION WITHOUT PACKET LOSS AND HOMOMORPHIC PROCESSING OF INTEGERS FOR SHARED TESTBED FACILITIES

Pedro Silveira Pisa

August/2013

Advisors: Otto Carlos Muniz Bandeira Duarte
Michel Abdalla

Department: Electrical Engineering

This master thesis addresses two fundamental issues for the Future Internet and shared testbed facilities: management of virtualized networks and processing of encrypted data through homomorphic encryption.

Virtual network technology increases network management complexity mainly due to increased monitoring variables and resource allocation possibilities. The first part of this work presents a tool for managing testbed facilities called VNEXT (Virtual NEtwork Management for Xen-based Testbeds), which was designed and developed by Grupo de Teleinformática e Automação (GTA). In the context of VNEXT, this master thesis aggregates a live migration mechanism without packet loss, an architectural reformulation of the VNEXT controller, and a graphical interface for network topology visual representation.

Homomorphic encryption, which is the subject of the second half of this work, is a method that allows processing encrypted data without decrypting the data in advance. This technique brings new privacy and security possibilities for data processing and storing in unreliable environments, such as shared testbed facilities and cloud computing. This work proposes an encryption scheme for larger integer numbers. The proposal is an extension of the DGHV [1] scheme. We implemented both the DGHV scheme and the proposed extension. The results show that our extension has better processing efficiency and similar memory usage. We also observe the key generation process is computationally expensive.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Símbolos

$(\mathbb{Z}_N)^*$	Conjunto dos números inteiros invertíveis na base N., p. 76
B	Base Utilizada no Esquema, p. 59–66, 69, 72, 73, 77, 78, 80–84, 87, 88
DEC	Algoritmo de Decriptação, p. 45, 46
$DEC^+_{\mathfrak{E}}$	Algoritmo de Decriptação Aumentado para Adição, p. 46
$DEC^\times_{\mathfrak{E}}$	Algoritmo de Decriptação Aumentado para Multiplicação, p. 46
$DEC_{\mathfrak{E}}$	Algoritmo de Decriptação Aumentado, p. 46
ENC	Algoritmo de Encriptação, p. 45, 46, 69
$EVAL$	Algoritmo de Avaliação, p. 45, 46
K_{priv}	Chave Privada, p. 40, 45–47, 54, 55, 57, 60–66, 69, 71, 80, 82, 83
K_{pub}	Chave Pública, p. 40, 45–47, 55, 61, 62, 83
$KeyGen$	Algoritmo de Geração de Chaves, p. 45, 46
N	Base qualquer utilizada na aritmética modular., p. 76
N	Valor máximo para o texto cifrado no RSA, p. 40
R	Ruído Adicionado ao Texto Cifrado, p. 55, 62–66
S	Conjunto de Elementos da Chave Pública usados para Encriptação, p. 55, 62–66
S_Φ	Conjunto de Circuitos Corretos para o Esquema, p. 45
S_Φ	Conjunto dos Circuitos Corretos para o Esquema, p. 45, 46
Φ	Circuito de Operações, p. 44–46

Φ Conjunto de Operações, p. 60

Ψ Texto Cifrado, p. 40, 45, 46, 55, 56, 60, 62, 63, 65, 66, 69

ℓ Tamanho da Chave Privada, p. 46

ϵ Vantagem do Atacante sobre um Esquema de Encriptação, p. 46, 71, 72

η Tamanho da Chave Privada, p. 54, 61, 71, 72

γ Tamanho dos Elementos da Chave Pública, p. 54, 55, 61, 71, 72

λ Parâmetro de Segurança, p. 46, 54, 57, 60, 61, 64–66, 71–73, 76–78, 80–84, 87, 88

$\mathbb{Z}$ Conjunto dos Números Inteiros, p. 63

$\mathcal{A}$ Algoritmo, p. 46, 72

$\mathcal{D}_{\gamma,\rho}(K_{priv})$ Distribuição Aleatória para Geração da Chave Pública, p. 55, 61

$\mathfrak{A}$ Atacante ao Esquema de Encriptação, p. 46, 72

$\mathfrak{E}$ Esquema de Encriptação, p. 45–47, 56, 69

$\mathfrak{E}^d$ Esquema de Encriptação Correto para Circuitos de Profundidade d, p. 46, 47

π Mensagem em Texto Claro, p. 40, 45, 46, 55, 62–66

ρ Tamanho do Ruído dos Elementos da Chave Pública, p. 54, 55, 61, 64, 71, 72

ρ' Tamanho do Ruído nos Textos Cifrados, p. 54, 61, 64, 71, 72

τ Quantidade de Elementos na Chave Pública, p. 54, 55, 61, 64, 71, 72, 83

$\vec{\Psi}$ Vetor de Textos Cifrados, p. 45, 46

a Número qualquer em aritmética modular., p. 76

d Profundidade do Circuito, p. 46, 47, 65, 66

d Profunfidade do Circuito, p. 46

g Base qualquer na operação de exponenciação., p. 75

k Quociente Inteiro da Divisão Modular, p. 63–65

n Número de mensagens e textos cifrados em operação, p. 45, 46

q_i Número Inteiro Gerador do Elemento i da Chave Pública, p. 55, 57, 61, 63, 71, 83

r_i Ruído Adicionado ao Elemento i da Chave Pública, p. 55, 61–65, 71, 83

x Expoente qualquer na operação de exponenciação., p. 75

x_i Elemento de Ordem i da Chave Pública, p. 55, 57, 61–66, 71, 83

Lista de Abreviaturas

3DES	*Triple Data Encryption Standard*, p. 4
AES	*Advanced Encryption Standard*, p. 4, 40, 86
ARP	*Address Resolution Protocol*, p. 31
AS	*Autonomous System*, p. 1
CABO	*Concurrent Architectures are Better than One*, p. 12
CCA2	*Adaptative Chosen Ciphertext Attack*, p. 58
CCA2	*Adaptive Chosen Ciphertext Attack*, p. 42
CCA	*Chosen Ciphertext Attack*, p. 42, 52
CIA	*Central Intelligence Agency*, p. 10
CPA	*Chosen Plaintext Attack*, p. 42, 47
CPU	*Central Processing Unit*, p. 26–28, 30
DDH	*Decisional Diffie–Hellman*, p. 48
DGHV	Dijk, Gentry, Halevi e Vaikuntanathan, p. 40, 59–62, 69–73, 77, 78, 80–84, 87, 88
DHCP	*Dynamic Host Configuration Protocol*, p. 10
DNS	*Domain Name Service*, p. 10
DoS	*Denial of Service*, p. 19
GMP	*The GNU Multiple Precision Arithmetic Library*, p. 73
GM	Goldwasser e Micali, p. 49
GRE	*Generic Routing Encapsulation*, p. 17
GTA	Grupo de Teleinformática e Automação, p. 23, 27, 85

GUI	*Graphical User Interface*, p. 26, 27
HIP	*Host Identity Protocol*, p. 9
HTTP	*HyperText Transfer Protocol*, p. 25, 26
IP	*Internet Protocol*, p. 2, 8, 27
ISP	*Internet Service Providers*, p. 1, 12
JSON	*JavaScript Object Notation*, p. 25
LWE	*Learning With Errors*, p. 57
MAC	*Media Access Control*, p. 31
MDC	Máximo Divisor Comum, p. 71, 72
MIB	*Management Information Bases*, p. 18
MLN	*Manage Large Networks*, p. 23
MPFR	*Multiple Precision Floating-Point Reliably*, p. 73
MV	Máquina Virtual, p. 30, 31
NAT	*Network Address Translation*, p. 10
NOC	*Network Operations Centers*, p. 18
P2P	*Peer to Peer*, p. 9
PIR	*Personal Information Retrieval*, p. 52
PPTA	*Probabilistic, Polynomial-Time Algorithm*, p. 41
RLWE	*Ring Learning With Errors*, p. 57
RSA	Rivest, Shamir e Adleman, p. 4, 40, 41, 70, 86
SNMP	*Simple Network Management Protocol*, p. 9
TCP	*Transmission Control Protocol*, p. 2, 8, 13, 25, 86
UDP	*User Datagram Protocol*, p. 33, 34
UML	*User Mode Linux*, p. 23
VMM	*Virtual Machine Monitor*, p. 30

VNEXT *Virtual NEtwork Management for Xen-based Testbeds*, p. 3, 4, 23–28, 85, 86

VPN *Virtual Private Networks*, p. 16, 17

WWW *World Wide Web*, p. 9

XML *Extensible Markup Language*, p. 25

XOR Operação de OU Exclusivo, p. 49

Capítulo 1

Introdução

Com mais de dois bilhões de usuários[1], a Internet é um grande sucesso. Atualmente, os mais diversos equipamentos se conectam com a Internet através das redes sem fio, como as redes celulares e as conexões Wi-Fi. A população acessa a Internet em dispositivos como carros, casas inteligentes, geladeiras, portas retratos, televisões e muitos outros. No entanto, a Internet é vítima do seu próprio sucesso, pois o seu crescimento e a dificuldade da implantação de inovações no núcleo da rede ameaçam a sua evolução. Devido ao seu tamanho, não foi mais possível o gerenciamento por apenas uma entidade. E, para solucionar o problema, foram criados diversos Sistemas Autonômos (AS - *Autonomous Systems*) e Provedores de Serviços de Internet (ISP - *Internet Service Providers*). No entanto, por sua natureza comercial, os ISPs não são adeptos a mudanças em seus equipamentos nem em sua arquitetura. Um dos principais argumentos contra a inclusão de novidades na Internet é a necessidade de testes dessas novidades, pois um problema causado por uma inovação que não foi cuidadosamente testada pode ter um impacto catastrófico, devido à importância que a Internet tem na economia e na sociedade mundial. Por outro lado, experimentar uma nova tecnologia em um cenário da escala da Internet é uma tarefa muito difícil tanto para um grupo de pesquisa quanto para uma grande empresa. Por esse motivo, os testes precisam ser realizados e a tecnologia em questão homologada em uma instalação largamente utilizada e de abrangência global.

Embora seja possível organizar uma estrutura de testes de grandes proporções para as inovações na Internet, a sua arquitetura simples e distribuída ainda age como uma barreira para certos tipos de inovação. Um exemplo é a construção de mecanismos que garantam uma qualidade de serviço fim-a-fim para o usuário. Como cada pedaço da rede é administrado por um provedor diferente, é inviável criar mecanismos de alocação de banda ou garantia de latência em todo o caminho do fluxo de dados. Mesmo que fosse possível tecnicamente, a auditoria para verificar se o

[1] `https://www.cia.gov/library/publications/the-world-factbook/geos/xx.html`

problema foi resolvido é praticamente impossível. Outro exemplo de restrição provocada pela arquitetura da Internet atual consiste no bloqueio de usuários maliciosos na rede. Não é possível determinar *a priori* se um usuário é malicioso ou não [2]. Assim, os desafios da nova Internet abrangem áreas como a segurança, o acesso móvel, o roteamento, o endereçamento e a gerência da rede. Algumas propostas para solucionar alguns desafios são incompatíveis com o modelo TCP/IP adotado ou não resolvem o problema por completo [3].

A Internet foi projetada na década de 1970 para uso em instituições militares e de ensino. Nesses mais de quarenta anos, no entanto, o propósito da Internet mudou radicalmente, sobretudo o perfil dos seus usuários. No começo, estes eram usuários altamente especializados, que a utilizavam para a troca de mensagens. Atualmente, são usuários com os mais diversos objetivos e níveis de conhecimento possíveis, distribuídos por todo o planeta, favorecendo ataques contra os usuários com pouco conhecimento, como os SPAMs [4] e as páginas falsas (*phishing*) [5]. Tal heterogeneidade cria um ambiente repleto de conflitos [6]. Juntamente com o crescimento da Internet, diversas aplicações surgem na rede. Essas novas aplicações trazem grandes modificações na rede, como é o caso das redes par-a-par [7] e da computação em nuvem [8]. Um exemplo de problema agravado com a computação em nuvem é a segurança e a privacidade dos dados armazenados e processados nas nuvens. Como os ambientes são compartilhados, falhas na infraestrutura da rede podem expor dados sigilosos dos usuários. Devido a todos esses problemas, diversos projetos apontam para a construção de uma nova Internet, a Internet do Futuro [9], que deve ser flexível o suficiente para promover a implantação e o teste de novas aplicações no núcleo da rede, enquanto engloba a conexão com diversas redes de novas gerações existentes, como, por exemplo, as redes veiculares [10], as redes *ad-hoc* [11, 12], as redes de sensores [13] e as redes tolerantes a atrasos e conexões [14], e redes que ainda não foram propostas.

1.1 Motivação e Objetivos

Esta dissertação aborda dois temas principais para a inovação na Internet. O primeiro tema é a gerência de redes virtualizadas e de plataformas de testes. As principais propostas para a arquitetura da Internet do Futuro são baseadas na virtualização, que é uma tecnologia que possibilita o compartilhamento da infraestrutura de equipamentos entre diversas redes virtuais, que, por sua vez, podem utilizar sistemas operacionais e pilhas de protocolos distintas e até incompatíveis [15]. No entanto, qualquer proposta para a implantação na Internet como um todo requer testes exaustivos em diversos cenários, topologias e quantidade de elementos de rede, além de utilizar tráfego de usuários reais. Torna-se cada vez mais difícil definir um

modelo para o comportamento dos usuários na Internet e recriar em simuladores a escala da rede. Dessa forma, plataformas de testes de escala global são fundamentais para a validação de propostas e a sua implantação na Internet. A principal forma, porém, de construir uma infraestrutura de testes global, flexível e escalável é através da virtualização. Assim, mecanismos para gerenciar e controlar essa infraestrutura de rede virtualizada são essenciais para que os administradores da rede possam tomar as decisões corretas.

Na primeira parte dessa dissertação, apresenta-se a ferramenta VNEXT (*Virtual NEtwork Management for Xen-based Testbeds*), que é um esforço coletivo do Grupo de Teleinformática e Automação (GTA) para criar uma infraestrutura de testes baseada em virtualização. A ferramenta utiliza o Xen [16], que é uma plataforma de virtualização para computadores pessoais de código aberto. Essa dissertação apresenta e propõe algumas soluções que são incorporadas à ferramenta VNEXT, como um mecanismo de migração de roteadores virtuais sem perdas de pacotes de rede, a arquitetura do controlador da ferramenta e uma interface de visualização da topologia das redes física e virtuais.

A principal proposta da parte I da dissertação é o mecanismo de migração sem perda, fundamental para a reorganização da topologia de rede virtual sobre a rede física. Uma das grandes vantagens da virtualização de redes é a realocação dinâmica dos recursos virtuais sobre os recursos físicos, pois permite, entre outras aplicações, a economia de energia, a manutenção preventiva e o balanceamento de carga. Com a migração ao vivo dos roteadores virtuais, essas aplicações são obtidas sem que a rede seja interrompida em nenhum momento.

A plataforma Xen, sobre a qual o VNEXT é baseado, possui um mecanismo de migração nativo. No entanto, esse mecanismo foi projetado para a migração de servidores não é indicado para a migração de roteadores, uma vez que pacotes são perdidos durante o processo de migração da máquina virtual. Essa perda de pacotes ocorre, pois a máquina virutal é suspensa por um curto período de tempo durante a migração. No cenário de servidores virtuais, as aplicações são protegidas por algoritmos de entrega confiável de perdas de pacotes. Esses algoritmos podem ser executados na camada de transporte ou dentro da própria aplicação. Por outro lado, em redes virtuais, as perdas de pacotes são graves, já que a camada de rede não está protegida por nenhum mecanismo de recuperação de pacotes perdidos.

Dessa forma, o mecanismo de migração ao vivo proposto não possui perda de pacotes, pois faz com que o encaminhamento de pacotes pelo roteador se mantenha funcional durante toda a migração do roteador virtual. Isso é obtido com a técnica de separação de planos, proposta por Wang et al. [17], na qual o encaminhamento de pacotes do roteador virtual é executado no roteador físico. Assim, mesmo com a interrupção do funcionamento da máquina virtual, os pacotes continuam a ser

encaminhados.

A ferramenta VNEXT consiste, de maneira geral, em um ambiente para testes de propostas para a Internet do Futuro. Através dela, os pesquisadores participantes podem implementar, instalar, testar e coletar medidas de suas propostas. Esse é um cenário compartilhado e, dessa forma, a ferramenta VNEXT requer mecanismos de segurança e privacidade para os dados.

Relacionado a privacidade dos dados, no cenário de computação em ambiente compartilhado, como na computação em nuvem e em plataformas compartilhadas de testes, existem dois problemas principais: o armazenamento dos dados e o processamento dos dados. O armazenamento dos dados é um problema conhecido e diversas soluções práticas existem, tanto ligadas à criptografia simétrica, como os algoritmos 3DES (*Triple Data Encryption Standard*) e AES (*Advanced Encryption Standard*), como ligadas à criptografia assimétrica, como diversas implementações do RSA.

O principal problema do ambiente compartilhado e a criptografia reside no processamento dos dados, pois na criptografia tradicional, os dados não podem ser alterados enquanto estão encriptados. É o conceito da não-maleabilidade e requer que todos os dados sejam descriptografados antes de serem processados, mesmo em ambiente compartilhado. Dessa forma, a segurança dos dados está comprometida. No entanto, em 1978, Rivest, Adleman e Dertouzos [18] apresentaram um conceito para o processamento de dados encriptados, os homomorfismos secretos. Atualmente, esse campo de pesquisa é intitulado de Criptografia Homomórfica e, na segunda parte dessa dissertação, aborda-se o tema, incluindo as principais propostas e desafios, bem como se propõe um esquema de encriptação apropriado para a realização de operações aritméticas de números inteiros grandes.

Entre as aplicações práticas da encriptação homomórfica, pode-se citar:

Computação multiparte Na computação multiparte, calcula-se uma função f a partir de contribuições de diversas entidades. Com a encriptação homomórfica, pode-se realizar essa computação sem que os parâmetros passem a ser conhecidos por quem não os conheciam antes da computação da função f. Assim, diversas entidades, como pessoas, empresas ou até mesmo governos podem contribuir com informações secretas para determinado cálculo com a garantia que as outras entidades que participam dessa computação não terão qualquer informação sobre os dados enviados.

Busca sobre informações privadas Com a criptografia homomórfica, pode-se construir uma base de dados em que os dados são armazenados encriptados. Assim, para a recuperação parcial desses dados, realiza-se uma operação algébrica entre os dados encriptados e os critérios de busca, que também devem

ser encriptados. Assim, pode-se armazenar os dados em um ambiente inseguro e recuperá-los com segurança.

Votações eletrônicas A criação de um sistema eletrônico de votação ainda é um desafio, mas, com a criptografia homomórfica, pode-se criar um mecanismo de contagem de votos sem que seja necessário o armazenamento em claro dos votos dos eleitores. Para essa aplicação, basta um esquema de encriptação homomórfico para a operação de adição, pois apenas deseja-se somar 1 a um montante de votos já computados.

Cálculos financeiros Extratos bancários e portfólios de investimentos dos clientes de instituições financeiras são informações privadas. Devido a isso, os bancos e corretoras mantêm toda uma infraestrutura própria para o armazenamento e o processamento dessas informações. Com a criptografia homomórfica, pode-se utilizar a computação nuvem para processar esses dados de maneira segura e garantir a privacidade dos clientes.

Diversas propostas de criptografia homomórfica já foram anunciadas e, hoje, sabe-se que é possível criar um mecanismo de processamento genérico dos dados encriptados. A proposta de Craig Gentry [19] provou a que é possível criar um mecanismo capaz de processar funções de adição e multiplicação binárias. No entanto, o esquema de encriptação proposto por Gentry não é prático do ponto de vista de armazenamento e de processamento. Existem outras propostas com o objetivo de simplificar [1] ou otimizar alguns aspectos da proposta de Gentry [20], mas elas ainda não são práticas.

Nessa dissertação, apresenta-se um esquema de criptografia homomórfica capaz de processar funções aritméticas com números inteiros grandes. Ao contrário das propostas existentes, cujo processamento é feito bit a bit, a proposta apresentada na dissertação visa o processamento dos números inteiros com uma única operação. Para armazenar e processar um número em outras propostas, precisa-se criptografar cada bit separadamente. Com o esquema proposto, os números são criptografados como números inteiros e processados com uma única operação de adição ou multiplicação. Com esse menor número de operações, objetiva-se um melhor desempenho global do esquema de encriptação homomórfica.

1.2 Organização da Dissertação

Essa dissertação é organizada da seguinte forma. No Capítulo 2, são discutidos conceitos e desafios das redes virtualizadas, bem como propostas de arquitetura para a Internet do Futuro. No Capítulo 3, apresenta-se a ferramenta VNEXT, assim

como algumas propostas dessa dissertação que foram incluídas na ferramenta, como a migração com separação de planos, culminando numa discussão sobre a segurança dos dados em plataformas de testes compartilhadas. No Capítulo 4, discute-se os príncipios de segurança e os conceitos da encriptação homomórfica, suas principais propostas e seus desafios de implementação. O Capítulo 5 apresenta uma proposta de esquema de criptografia homomórfica para realizar operações aritméticas com números inteiros grandes e evidencia análises de corretude do esquema e de determinação do máximo de operações que podem ser executadas com o esquema proposto em função dos seus parâmetros, além de uma discussão sobre as bases de segurança do esquema, dos detalhes de sua implementação e uma análise experimental do esquema de encriptação com relação ao seu desempenho e sobrecarga de armazenamento. E, finalmente, as conclusões são apresentadas no Capítulo 6.

Parte I

Gerenciamento de Redes Virtualizadas

Capítulo 2

Redes Virtualizadas

O modelo de núcleo simples e inteligência nas extremidades e os protocolos TCP/IP foram os maiores responsáveis pelo grande sucesso da Internet. No entanto, esse mesmo modelo engessa o núcleo da rede, pois não é possível configurar ou inserir novos elementos no núcleo da rede [21]. Além disso, a baixa capacidade de gerenciamento da rede implica configurações manuais, difícil depuração de erros e barreiras na implantação de novas tecnologias. Há um consenso na comunidade científica sobre a necessidade de uma nova Internet [3], a Internet do Futuro. Uma proposta para essa nova Internet é baseada na tecnologia de virtualização, que permite diversas redes virtuais, executando protocolos distintos, funcionarem ao mesmo tempo e de forma isolada sobre o mesmo substrato físico [22, 23]. A manutenção dessa infraestrutura de redes virtuais se configura como um desafio para o gerenciamento de redes, visto que é necessário considerar as especificidades de cada rede, a crescente quantidade de dados coletados, os diferentes interesses de cada proprietário das redes virtuais sobre a arquitetura e, sobretudo, as restrições de qualidade de serviço e isolamento de cada rede [24].

2.1 Conceitos sobre a Internet e a Virtualização

A Internet é um projeto da década de 70 que surgiu a partir de redes militares e universitárias. Tinha por objetivo principal o envio de mensagens eletrônicas entre as entidades participantes da rede, constituídas por usuários altamente especializados. Em seu projeto original, a prioridade foi dada sobre a simplicidade no núcleo da rede, que representa os elementos intermediários necessários para prover a conectividade entre os sistemas finais, responsáveis pelas aplicações que utilizavam a rede. Dessa forma, toda a inteligência da rede se localiza nas extremidades e o núcleo da rede ficou responsável pelo encaminhamento dos pacotes transmitidos, semelhante a um sistema de correio convencional. O protocolo responsável pelo endereçamento dos pacotes é utilizado até hoje e é conhecido como *Internet Protocol* (IP). Esse modelo

possibilita a fácil implantação de novas aplicações e novos protocolos na Internet, pois todo o núcleo da rede permanece inalterado, havendo apenas necessidade que os sistemas finais, nas extremidades da rede, recebam a implementação dessas novas aplicações. Tal característica permitiu o rápido crescimento de aplicações como a *World Wide Web* (WWW) e as redes Par-a-Par (P2P), que se mostraram um sucesso ao longo das últimas décadas.

No entanto, a maior vantagem da Internet é também a sua principal desvantagem. Diversas aplicações, como as que envolvem a mobilidade e a correção de falhas, poderiam se aproveitar de mecanismos inteligentes no núcleo da Internet, caso eles existissem, para melhorar seu desempenho. A restrição para a mobilidade reside na sobrecarga semântica do endereço IP, que acumula as funções de identificador e localizador. Dessa forma, um endereço IP está associado a uma localização geográfica, sendo necessária a sua mudança quando o usuário se move. Ao mudar o endereço IP, as conexões criadas entre as extremidades se perdem. Sistemas inteligentes e capazes de realizar o procedimento de *handover* auxiliariam na manutenção da comunicação de forma transparente para o usuário e para a rede. Protocolos como o *Host Identity Protocol* (HIP) [25, 26] visam remover essa sobrecarga semântica permitindo a mobilidade dos usuários. Com o objetivo de desacoplar definitivamente a localização da informação buscada, surgem as Redes Centradas em Conteúdo (*Content-Concentric Networks* - CCN) [27], um novo paradigma para a Internet em que o foco é a entrega do conteúdo para os usuários independentemente da localização desse conteúdo, ao contrário da arquitetura atual da Internet em que o foco é a comunicação entre sistemas finais. Com as CCNs, a infraestrutura de rede possui mecanismos para mitigar alguns problemas da Internet atual, como a necessidade de garantir a segurança do repositório de dados e caminho que os dados percorrem [28]. Por outro lado, essas redes criam novos problemas para a rede como, por exemplo, no roteamento, em que a quantidade de rotas e de mensagens de controle aumentam consideravelmente, criando novos desafios para os protocolos de roteamento [29].

As falhas que ocorrem na Internet não são corretamente relatadas, pois os elementos do núcleo da rede são simples e não informam sobre o funcionamento dos equipamentos vizinhos e da rede como um todo. Além da própria inexistência de registros das falhas, a Internet é fragmentada em sistemas autônomos, cada um responsável por gerir um pedaço da rede [21]. Essa fragmentação da gerência significa que as informações coletadas por um sistema autônomo podem não estar disponíveis para outros sistemas autônomos. Nesse cenário, mecanismos de diagnóstico de rede precisam ser inseridos nas extremidades para descobrir o real problema detectado. Na década de 80, o *Simple Network Management Protocol* (SNMP) [30] foi definido e se tornou o protocolo padrão para o gerenciamento da Internet. Atualmente, os equipamentos de rede implementam o protocolo SNMP, mas sua atuação é limitada

ao monitoramento dos equipamentos, sem atuar na gerência das configurações dos protocolos e aplicações utilizados nos equipamentos [31].

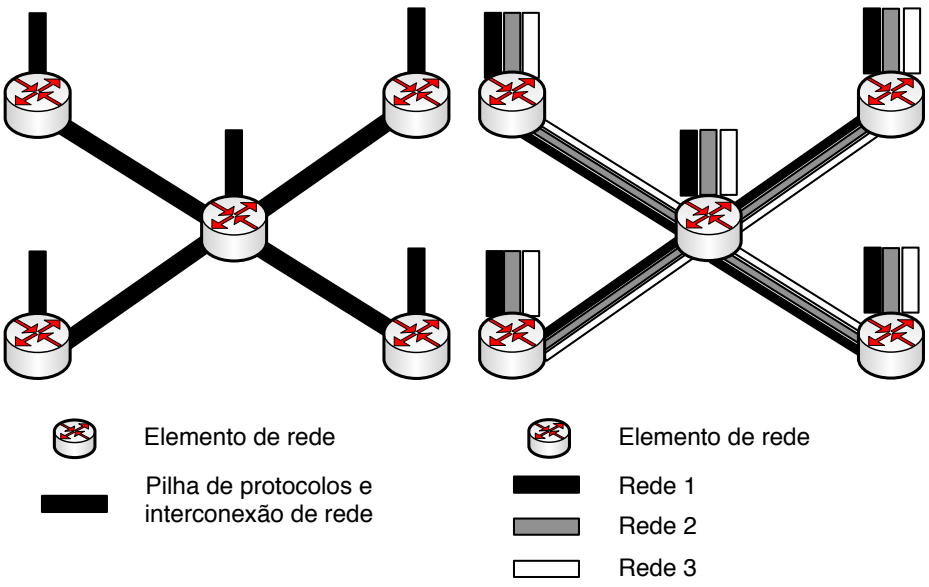
2.2 Modelos de Virtualização de Rede

Atualmente, a Internet se consagra com um grande sucesso. Segundo a Agência Central de Inteligência (*Central Intelligence Agency* - CIA) dos Estados Unidos, em 2010, existiam mais de 2 bilhões e 100 milhões de usuários de Internet no mundo[1]. O Brasil, em 2009, era o quarto país em número de usuários da Internet, com quase 76 milhões, atrás apenas da China, dos Estados Unidos e do Japão[2]. Com esse sucesso, o objetivo principal da Internet e o nível de especialização dos seus usuários mudaram radicalmente. Demandas por novas aplicações surgem constantemente e os requisitos que nortearam a criação da Internet são colocados à prova. A evolução para a rede que existe hoje ocorreu com base em "remendos" ao projeto original, como o *Network Address Translation* (NAT), o *Dynamic Host Configuration Protocol* (DHCP) e o *Domain Name Service* (DNS). NAT e DHCP são mecanismos criados para postergar a escassez de endereços IP disponíveis. NAT [32] permite que diversos dispositivos se conectem à Internet utilizando um mesmo endereço IP, quebrando premissas fundamentais, como o endereço IP global e único e a comunicação fim-a-fim, já que é necessário um elemento intermediário na rede para fazer a tradução do endereço interno para o endereço aparente na Internet. DHCP [33] é um mecanismo de alocação dinâmica de endereços e permite que dispositivos diferentes utilizem um mesmo endereço IP em momentos distintos, otimizando a distribuição dos endereços disponíveis para um dado domínio da rede. Por fim, DNS [34] é um protocolo de mapeamento criado para facilitar a memorização dos endereços da Internet na forma de nomes relacionados ao destino da mensagem.

Assim, existe uma tendência entre os pesquisadores de que a arquitetura da Internet precisa ser reformulada [3]. Considera-se que a arquitetura original da Internet a engessa, dificultando a inovação no núcleo da rede. Uma parte da comunidade científica defende que a reformulação seja completa, ignorando a compatibilidade com a Internet atual, como no projeto *Clean Slate* [9]. Dessa forma, uma nova Internet deve ser criada do zero, considerando seus os novos requisitos e mantendo os itens que garantiram seu sucesso, como a facilidade na implantação de novos serviços e a adaptabilidade dos protocolos da rede a esses novos serviços.

No contexto da concepção da nova Internet, pautada nos princípios que levaram ao sucesso a Internet atual, sem a preocupação com a compatibilidade e com novos conceitos como a autonomia, existem dois modelos principais para a construção da

[1] `https://www.cia.gov/library/publications/the-world-factbook/geos/xx.html`
[2] `https://www.cia.gov/library/publications/the-world-factbook/geos/br.html`



(a) Modelo purista. (b) Modelo pluralista.

Figura 2.1: Representação da estrutura de interconexão de rede e das pilhas de protocolos em cada um dos modelos de virtualização de rede.

arquitetura flexível da Internet do Futuro: o modelo purista e o modelo pluralista. As pilhas de protocolos e a interconexão entre os elementos de rede em cada um dos modelos é comparada na Figura 2.1.

2.2.1 O Modelo Purista

Os defensores do modelo purista acreditam que uma arquitetura para a Internet do Futuro deva possuir a flexibilidade em sua concepção. Essa arquitetura deve englobar todas as ferramentas necessárias para assegurar a integração de soluções específicas para diversos desafios de qualidade de serviço, roteamento e gerenciamento, entre outras. Atualmente, diversas propostas para problemas específicos da rede existem e são incompatíveis. No modelo purista, elas serão integradas na arquitetura da rede, que deve possuir uma pilha de protocolos integrada e interconexão única entre os elementos da rede, como apresentado na Figura 2.1(a). Dessa forma, para buscar a flexibilidade na arquitetura, os defensores do modelo purista não esperam impactos imediatos para os usuários da rede, pois a rede tem a capacidade de absorver gradualmente as inovações. Tecnologias como as redes sobrepostas (*overlays*) e a virtualização são encaradas como ferramentas para agregar as novas funcionalidades à arquitetura.

2.2.2 O Modelo Pluralista

Os defensores do modelo pluralista acreditam na coexistência de múltiplas pilhas de protocolo ao mesmo tempo na Internet, como ilustrado pela Figura 2.1(b). Nessa visão, técnicas como a virtualização de redes e as redes sobrepostas são fundamentais para a arquitetura. Assim, cada rede instanciada objetiva a solução de um problema específico, facilitando a implantação e permitindo a percepção imediata das inovações no núcleo da rede. Além disso, os pluralistas defendem que a arquitetura é mais dinâmica e permite evoluções maiores, já que não requer a integração e a compatibilidade das soluções propostas para problemas específicos, visto que cada uma das propostas é instanciada simultânea e isoladamente em redes sobrepostas ou multiplexadas por uma ferramenta de virtualização. Pode-se, portanto, imaginar um cenário extremo, em que para cada problema ou requisito existente, cria-se uma rede específica. No modelo pluralista, a migração para a nova Internet é automática, uma vez que a Internet atual pode ser instanciada e compor uma das pilhas de protocolos utilizadas.

A ferramenta VNEXT [35], detalhada no Capítulo 3 desta dissertação, é um esforço coletivo do Grupo de Teleinformática de Automação. Essa ferramenta é concebida para o modelo de redes pluralista, no qual as redes virtuais são parte da arquitetura da nova Internet. O modelo pluralista é adotado pela ferramenta devido às vantagens de implantação, evolução e coexistência de novas soluções. Assim, a ferramenta VNEXT, para a qual esta dissertação contribui com propostas relacionadas à migração de roteadores virtuais e visualização de topologia, é projetada para o gerenciamento de redes virtuais em ambiente pluralista.

2.3 Arquitetura de Redes Virtuais

A Internet atual é fragmentada em sistemas autônomos, que são gerenciados por Provedores de Serviços da Internet (ISPs - *Internet Service Providers*). Assim, os ISPs são responsáveis por prover tanto os serviços da rede quanto a infraestrutura de acesso. Além disso, os ISPs possuem gerência apenas sobre a sua rede, inviabilizando a oferta de serviços diferenciados, pois não detém o controle de todo o caminho fim-a-fim. Com o objetivo de flexibilizar este cenário, propostas surgem com o objetivo de separar os papéis acumulados pelo ISP. O projeto *Concurrent Architectures are Better than One* (CABO) [36] defende a utilização de roteadores virtuais, sob responsabilidade dos provedores de serviço, executando sobre os roteadores físicos, que, por sua vez, são geridos pelos provedores de infraestrutura. Sob o aspecto dos provedores de infraestrutura, continua sendo inviável um mesmo provedor deter o controle de todo o caminho fim-a-fim, devido aos aspectos técnicos,

sobretudo relacionados à escala da infraestrutura que seria necessária. No entanto, para os provedores de serviço, o controle de todo o caminho fim-a-fim é possível, uma vez que os roteadores virtuais são instâncias de software executadas sobre o substrato físico dos provedores de infraestrutura. Com essa arquitetura, cada provedor de serviço pode alterar a configuração de todos os seus equipamentos ou até mesmo utilizar uma pilha completamente nova de protocolos, sem que os demais provedores de serviços sejam afetados.

Outras propostas semelhantes são apresentadas, como é o caso do projeto 4WARD [37, 38], que defende a coexistência de três entidades: os provedores de infraestrutura, os provedores de redes virtuais e os operadores das redes virtuais. A organização é semelhante à proposta pelo projeto CABO, mas subdivide o provedor de serviço em duas entidades distintas. O provedor de redes virtuais é responsável por contratar redes virtuais e garantir os contratos com os provedores de infraestrutura. O operador de redes é a entidade usuária da rede e quem realmente proverá o serviço para o usuário da rede. O operador de redes é responsável por realizar as configurações e aperfeiçoamentos na rede para atender às necessidade do usuário. O elemento comum das propostas é a dissociação entre o responsável pela infraestrutura e o provedor de serviço através da virtualização de redes. Nesse contexto, a virtualização recebe um importante papel na Internet do Futuro. Com ela, possibilita-se a concretização das propostas de arquitetura para a Internet, sobretudo no modelo pluralista. Assim, com a adoção da virtualização de redes, é possível *criar*, *instanciar*, *migrar* e *remover* roteadores virtuais sob demanda e com agilidade. Tais primitivas permitem agregar à rede uma série de características como a manutenção de equipamentos físicos sem quedas nos serviços, economia de energia, garantia de qualidade de serviço, dentre outras. Esta parte da dissertação se concentra na gerência de redes virtualizadas e no modelo pluralista para a Internet. Mais especificamente, os roteadores virtuais são desenvolvidos sobre a plataforma de virtualização Xen [16], que foi projetada para a virtualização de servidores em *datacenters*. Por esse motivo, o Xen utiliza a plataforma de computadores pessoais padrão de mercado, com arquitetura x86 e amd64, dentre outras. A adoção de uma plataforma de virtualização para computadores pessoais facilita a implantação, devido ao custo dos equipamentos, mas possui restrições de projeto, como é o caso da migração, projetada para aplicações que executam sobre o protocolo TCP, uma vez que causa a perda de pacotes de rede e o TCP pode recuperá-los.

2.4 Desafios de Gerenciamento de Redes Virtuais

Muitos trabalhos científicos existentes relacionados à virtualização de redes podem ser descritos como uma tentativa de consertar problemas existentes ao invés

de serem vistos como um avanço consciente e focado para construir um ambiente virtual completo de redes [39]. Como resultado, diversos aspectos da virtualização de redes continuam inexplorados e muitos outros requerem modificações ou melhoramentos. Esta seção busca apresentar os principais desafios do gerenciamento de uma infraestrutura de redes virtuais sob diversos aspectos, como o fornecimento de uma interface de gerência, a topologia das redes, a alocação de recursos, as políticas de uso e controle de admissão, a construção dos enlaces e nós virtuais, o endereçamento e a identificação dos nós virtuais, o gerenciamento da mobilidade, o monitoramento, a configuração e o tratamento de falhas, questões de segurança e privacidade, problemas de interoperabilidade e o modelo de negócios da virtualização de redes.

Interfaces

Todo provedor de infraestrutura deve prover uma interface para os provedores de serviço expressarem seus requisitos. Além disso, essas interfaces devem ser padronizadas e permitir a programação dos elementos de rede disponíveis para os provedores de serviço. De forma similar, interfaces apropriadas entre os usuários das redes virtuais e os provedores de serviço, entre diversos provedores de infraestrutura e diversos provedores de serviço devem ser identificadas e padronizadas. Sem essas interfaces de comunicação, os elementos da rede não podem ser configurados e programados pelos provedores de serviço, impedindo e anulando a vantagem do aproveitamento da característica de programação das redes virtuais.

Sinalização e Inicialização da Rede Virtual

Antes da criação da rede virtual, o provedor de serviço precisa ter conectividade de rede com todos os provedores de infraestrutura a serem utilizados a fim de enviar suas requisições. Essa necessidade cria uma circularidade em que a conectividade de rede é um pré-requisito para si própria [40]. Devem existir também capacidades de inicialização para permitir que o provedor serviço personalize os nós e enlaces virtuais alocados para ele através de interfaces apropriadas. Ambos os requisitos obrigam a existência de pelo menos uma rede que deve estar sempre presente para prover a conectividade e possibilitar tratar essa etapa de configuração. O que significa que deve existir um mecanismo em uma rede separada para realizar a sinalização e inicialização da rede virtual.

Descoberta de Recursos e Topologia

Para alocar os recursos entre as requisições dos diversos provedores de serviços, os provedores de infraestrutura devem ser capazes de determinar a topologia das

redes que eles gerenciam assim como coletar as estatísticas de uso de cada um dos elementos da rede, incluindo os nós físicos e suas interconexões e as capacidades disponíveis nos nós e enlaces da rede. Além disso, dois provedores de infraestrutura adjacentes devem ser capazes de instanciar enlaces virtuais de borda, ou seja, entre os provedores de infraestrutura, para construir redes virtuais fim a fim. Por outro lado, sob o ponto de vista dos provedores de serviço, existem casos em que deve ser possível, a partir de uma rede virtual, descobrir a presença e a topologia de outras redes virtuais coexistentes, permitindo que as redes virtuais se comuniquem, interajam e colaborem entre si para prover serviços maiores e mais complexos. No entanto, para não comprometer as características de isolamento da arquitetura, essa comunicação e interação deve ser administrada por um conjunto de regras definidos entre os provedores de serviços.

Alocação de Recursos

A alocação e o agendamento eficiente dos recursos da rede física entre as múltiplas requisições de redes virtuais são extremamente importantes para maximizar o número de redes virtuais coexistindo simultaneamente no substrato físico, o que aumenta a utilização e o retorno financeiro para o provedor de infraestrutura. A alocação de recursos com restrições nos nós e enlaces virtuais, também conhecida como o problema de incorporação (*embedding problem*), é reconhecida como um problema NP-difícil. Soluções existentes baseadas em heurísticas dividem-se em dois grandes grupos para resolver o problema considerando um único provedor de infraestrutura. As soluções *offline* requerem o conhecimento das requisições dos provedores de serviço *a priori* e as soluções *online* calculam a melhor alocação possível conforme recebem as requisições dos provedores de serviço. Ambas as soluções são casos extremos e propostas híbridas também são apresentadas. Uma proposta para representar e resolver o problema de alocação de recursos nas redes virtuais, publicada por Chowdhury, Rahman e Boutaba [41], é baseada no problema dos inteiros mistos e outra proposta, baseada em colônias de formigas, por Fajjari, Aitsaadi, Pujolle e Zimmermann [42].

Embora as diversas restrições e objetivos transformem esse problema em um problema computacionalmente intratável, a presença de numerosas e diversificadas topologias e as possíveis oportunidades de explorar essas topologias ainda deixam espaço suficiente para a pesquisa em soluções personalizadas e melhores aproximações de algoritmos. Além disso, incorporar redes virtuais sobre o substrato físico de diversos provedores de infraestrutura é ainda um problema praticamente não abordado.

Políticas de Uso e Controle de Admissão

Quando se estabelece uma rede virtual, um provedor de serviço requer garantias específicas para os atributos de seus nós virtuais e características dos enlaces virtuais. Os provedores de infraestrutura devem realizar uma auditoria acurada e implementar algoritmos com políticas de controle de admissão e de distribuição dos recursos para garantir que os provedores de infraestrutura são capazes de prover a qualidade de serviço contratada e que as redes virtuais não excedam os recursos alocados local e globalmente. No entanto, novos algoritmos devem ser construídos para considerar a alocação de toda a rede virtual, ao invés do controle de admissão e dos algoritmos de políticas existentes, que são focados em nós e enlaces individuais, como no controlador VIPER [24]. No VIPER, um controlador é executado no Domínio 0 da máquina e gerencia o acesso aos recursos de rede por todas as máquinas virtuais além de possuir um controle de admissão de novos nós virtuais para impedir a saturação da utilização dos recursos.

Enlaces Virtuais e Nós Virtuais

Nós virtuais permitem que diversos provedores de serviços compartilhem o mesmo conjunto de recursos físicos e implementem protocolos de controle personalizados em cada nó. Atualmente, os fabricantes de roteadores oferecem nós virtuais como uma ferramenta para simplificar o projeto do núcleo da rede, diminuir os custos e com propósitos de criar redes privadas virtuais (VPN - *Virtual Private Networks*). Com a programação, esses conceitos podem ser estendidos para criar roteadores físicos que permitam a cada provedor de serviço a personalização de seus nós virtuais para atender às suas necessidades. Assim, a escalabilidade do ambiente de rede virtual como um todo está intimamente ligada com a escalabilidade dos elementos físicos usados pelos provedores de infraestrutura. O trabalho de Carvalho *et al.* [43] propõe um mecanismo *online* com heurística nebulosa para a alocação dos nós virtuais nos elementos físicos de acordo com o perfil de uso de cada elemento virtual, considerando os recursos de processamento, memória e tráfego de rede. No entanto, a proposta ainda é limitada no número de roteadores virtuais e de elementos físicos que podem ser controlados. Pesquisas nessa direção devem focar no número crescente de nós virtuais que os elementos físicos podem comportar.

Para construir a virtualização da rede, os enlaces entre os nós virtuais também precisam ser virtualizados. A funcionalidade de construir túneis sobre múltiplos enlaces físicos já está presente no contexto das VPNs. Mecanismos similares de tunelamento podem ser usados no caso das redes virtuais. No entanto, uma restrição que deve ser observada é que a velocidade de transferência de pacotes, considerando a largura de banda e a latência, nos enlaces virtuais deve ser comparável

com a velocidade observada nos enlaces físicos, o que significa uma sobrecarga de custo mínima relacionada ao encapsulamento e à multiplexação. Mattos *et al.* [44] propõem a utilização de comutadores OpenFlow [45] para o encaminhamento dos quadros na rede física. Para possibilitar a conectividade sobre a Internet, os autores utilizam a ferramenta *Capsulator* sobre um túnel *Generic Routing Encapsulation* (GRE). A ferramenta *Capsulator* cria um *proxy* entre duas redes locais distintas e o túnel GRE é utilizado para criar uma conexão fim a fim entre duas redes construídas nas universidades participantes do projeto piloto descrito na proposta pelos autores. Para aumentar a segurança da rede proposta, os autores constroem uma VPN criptografada, sobre a qual o túnel GRE é construído. Essa proposta permite a interconexão em nível de enlace entre diversas redes físicas e, através dessas interconexões, pode-se criar os enlaces virtuais. No entanto, devido à sobrecarga de cabeçalho adicionada pela proposta, a velocidade de transmissão de pacotes é reduzida.

Endereçamento e Identificação dos Nós Virtuais

O mapeamento entre os diferentes contextos de endereços é um problema bem conhecido na literatura quando se discute a multiplexação de uma rede física entre diversas redes, como é o caso da virtualização de redes. Quando se considera, por exemplo, a presença de diferentes e, às vezes, incompatíveis requisitos de endereçamento em cada uma das diferentes redes virtuais, esse mapeamento se torna ainda mais complexo [46].

A atribuição dos nomes e dos endereços deve estar desacoplada do ambiente de redes virtuais para que cada usuário final possa mudar de provedor de serviços preservando a sua identidade única. Nesse contexto, o usuário final pode se conectar simultaneamente a diversas redes virtuais de diferentes provedores de serviços. Embora esse conceito pareça semelhante ao *multihoming*, o problema é aumentado pela possível heterogeneidade das diferentes redes virtuais [46].

Gerenciamento da Mobilidade

Em um ambiente de redes virtuais, a mobilidade dos dispositivos deve ser tratada como uma premissa básica, sem que haja a utilização de soluções improvisadas ou "remendos", como ocorre na Internet atual. Mobilidade, nesse contexto, significa não somente a simples movimentação geográfica dos dispositivos dos usuários finais, mas também as movimentações dos elementos virtuais do núcleo da rede, as quais podem ser realizadas através da técnica de migração. Um desafio a mais nesse contexto é que as plataformas de virtualização existentes não estão preparadas para a migração de roteadores, pois os algoritmos de migração ao vivo possuem um período

em que o roteador é desligado. Por menor que seja esse período, nesse momento, o roteador não recebe e encaminha os pacotes em trânsito, perdendo-os de maneira irreversível[3]. Pisa *et al.* [47] abordaram este problema e desenvolveram uma solução através da separação de planos de controle e de dados dos roteadores inicialmente proposta em VROOM [17]. O plano de controle consiste nos algoritmos de roteamento responsáveis pela comunicação com os nós vizinhos e pelo cálculo das rotas. Já o plano de dados contém a tabela de encaminhamento e é utilizado para o encaminhamento dos pacotes de rede propriamente ditos. Com a separação dos planos, o plano de controle continua no elemento virtual, dando liberdade para o usuário da rede utilizar o algoritmo que desejar, enquanto o plano de dados é transferido para o elemento físico e, por isso, permanece ligado durante toda a migração, evitando, assim, a perda de pacotes no processo. Essa técnica é incorporada na ferramenta VNEXT [35], que é apresentada em detalhes, juntamente com a técnica de migração proposta no Capítulo 3. No entanto, essa arquitetura reduz a flexibilidade da rede, pois todos os mecanismos de encaminhamento desejados devem ser implementados no elemento físico, o que pode inviabilizar tecnicamente novas propostas a serem testadas. Além disso, com a separação de planos, os recursos do elemento físicos são compartilhados, reduzindo as garantias de qualidade de serviço. Considerando essa movimentação dos dispositivos, encontrar a correta localização de qualquer dispositivo em um momento particular e rotear os pacotes de acordo com essa localização é uma tarefa difícil. Além disso, usuários finais podem se mover logicamente de uma rede virtual para outra com o objetivo de acessar os diferentes serviços providos por cada uma delas, o que também aumenta a complexidade do problema.

Monitoramento, Configuração e Tratamento de Falhas

Para permitir que cada provedor de serviço possa configurar, monitorar e controlar as suas redes virtuais independentemente dos outros provedores, mudanças consideráveis são necessárias nos Centros de Operação de Redes (NOC - *Network Operation Centers*) para incluir agentes inteligentes nos elementos de rede de baixo nível [39]. O conceito de MIBLets [48] consiste em Bases de Informações de Gerenciamento (MIB - *Management Information Bases*) particionadas para coletar e processar estatísticas de desempenho para cada uma das redes virtuais existentes na infraestrutura. No entanto, um arcabouço robusto de monitoramento requer mais atenção e esforço.

A existência de falhas nos componentes da rede física pode provocar uma série de falhas em cascata nas redes virtuais que estão diretamente alocadas nos elementos

[3] A Seção 3.3 explica em mais detalhes o conceito e o funcionamento da migração de máquinas e roteadores em ambientes virtualizados.

falhos da rede física. Detecção, propagação e isolamento das falhas, assim como a proteção e a recuperação das falhas, são desafios de pesquisa em aberto.

Segurança e Privacidade

O isolamento entre as redes virtuais coexistentes em uma mesma rede física provê certo nível de segurança e privacidade através do uso de túneis seguros e encriptações. Contudo, isso não elimina as ameaças, invasões e ataques contra as redes física e virtuais, como os ataques de negação de serviço (DoS - *Denial of Service*) e invasões que exploram vulnerabilidades para utilizar a máquina como robô ou coletar informações. Além disso, questões específicas sobre a segurança e a privacidade de redes virtualizadas devem ser identificadas e exploradas. Por exemplo, a programação dos elementos de rede pode aumentar a vulnerabilidade se modelos e interfaces não estiverem disponíveis. Todos esses questionamentos requerem um exame apurado para a criação de ambientes realísticos de redes virtualizadas.

Problemas de Interoperabilidade

Redes virtuais fim a fim podem ser divididas entre múltiplos domínios administrativos e cada um deles pode utilizar tecnologias de rede e arcabouços de gerenciamento diferentes. Habilitar o uso da virtualização em cada uma dessas tecnologias requer soluções específicas para o fornecimento dos recursos, operação da rede e manutenção dos elementos físicos e virtuais. Possibilitar interações entre infraestruturas de rede distintas, enquanto oferece uma interface de gerenciamento genérica e transparente, para que os provedores de serviço possam facilmente compor e gerenciar as suas redes virtuais, continua sendo uma tarefa difícil. Além disso, a identificação da necessidade, do escopo e das interfaces necessárias para a comunicação fim-a-fim entre múltiplas redes virtuais merece um exame minucioso.

Modelos de Negócios da Virtualização de Redes

Ao contrário das redes tradicionais, em que a largura de banda é a medida de cobrança principal, os nós virtuais são tão importantes quanto os enlaces virtuais em um ambiente de redes virtuais. Os provedores de serviços são compradores nesse mercado e os provedores de infraestrutura, os vendedores. Podem existir também os corretores, que agem como mediadores entre os compradores e os vendedores. Os usuários finais podem participar como compradores de serviços em diferentes provedores de serviços.

Existem dois tipos de modelos: centralizado e descentralizado. Os modelos centralizados são eficientes, mas também vulneráveis e não escaláveis. Por outro lado,

os modelos totalmente descentralizados são extensíveis e tolerantes a falhas, mas são propensos ao comportamento malicioso e à ineficiência. Para encontrar uma solução ideal entre essas duas opções, que se torne viável para a implantação na Internet do Futuro, muito esforço em pesquisa deve ser realizado.

2.5 Propostas de Gerenciamento de Redes Virtuais

O emprego de técnicas de virtualização no cenário de redes permite a execução de múltiplas redes virtuais, chamadas fatias (*slices*), sobre o mesmo substrato físico [49]. As redes virtuais são isoladas e cada uma pode utilizar sua própria pilha de protocolos. Como a virtualização oferece a flexibilidade necessária para a Internet do Futuro e também permite que os protocolos da pilha TCP/IP sejam executados nas fatias, a virtualização tem sido usada como a base para plataformas de testes de redes do futuro [50, 51].

Redes virtuais são implementadas com a mesma abordagem utilizada para a consolidação de servidores. Similarmente, cada roteador físico é compartilhado entre diversos roteadores virtuais. Uma rede virtual é, portanto, um conjunto de roteadores virtuais e seus respectivos enlaces [52, 53]. A virtualização de roteadores pode ser realizada com o Xen [16]. Egi *et al.* [52] mostram como os computadores pessoais com múltiplos núcleos podem representar uma alternativa de baixo custo e com desempenho similar em relação aos roteadores vendidos no mercado. Além disso, enquanto os roteadores do mercado possuem apenas um plano de controle e um plano de dados, o Xen possibilita a virtualização de ambos os planos. O plano de controle é responsável pelo protocolo de roteamento e cálculo das rotas, enquanto que o plano de dados serve para o encaminhamento dos pacotes a serem transmitidos pelo roteador. Essa característica de virtualização dos planos de controle e de dados permite que cada roteador virtual tenha seu próprio plano de dados e de controle, melhorando a programação da rede. Esta é a maior vantagem do uso do Xen na virtualização de redes e, por isso, essa plataforma é utilizada em diversas instalações de testes para a Internet do Futuro e em propostas de novas arquiteturas para a Internet [50, 51].

Na virtualização de redes, a sobrecarga de gerenciamento das inúmeras redes virtuais transforma o gerenciamento em um desafio ainda maior do que o encontrado na Internet atual [54]. Alocar os recursos físicos entre as redes virtuais e também gerenciar e controlar os recursos usados por múltiplas redes virtuais são tarefas complexas [42, 55]. Nesse cenário, os administradores de rede precisam da assistência de sistemas de gerenciamento e controle alimentados com funcionalidades de tomada

de decisão. O número de parâmetros de rede monitorados e a influência de uma rede virtual no desempenho das demais redes virtuais aumentam consideravelmente a importância desses sistemas de gerência inteligente da rede. Além disso, existem novas funcionalidades que representam possibilidades adicionais ainda não integradas aos sistemas de gerenciamento de redes virtuais, como instanciação, migração e desligamento de um roteador virtual [23] ou de toda a rede virtual.

Capítulo 3

Controle e Gerenciamento de Redes Virtuais baseadas em Xen

Este capítulo apresenta uma ferramenta de controle e gerenciamento de redes virtuais baseada em Xen, a ferramenta VNEXT (*Virtual NEtwork management for Xen-based Testbeds*). A ferramenta VNEXT[1] é um trabalho coletivo do Grupo de Teleinformática e Automação que foi desenvolvida no Projeto Horizon[2] de cooperação internacional. O VNEXT é focado no gerenciamento e no controle de redes e roteadores virtuais e, portanto, é diferente dos sistemas existentes de gerenciamento também baseados na plataforma Xen. Existem algumas ferramentas de controle e gerenciamento de plataformas de virtualização de máquinas que possuem funções como a instanciação, o desligamento e a migração de máquinas virtuais. Estas propostas são adequadas para a consolidação de servidores e não possuem facilidade relacionadas a redes de computadores. Algumas propostas como, por exemplo, a ferramenta HyperVM [56] é projetada para gerenciar um grupo de servidores virtualizados, provendo uma interface gráfica para a Web e outras funções como *traffic shaping* e gerenciamento de redes. Apesar de usar ferramentas de gerenciamento para um ambiente de rede virtualizado, o HyperVM considera a máquina virtual como um roteador e, consequentemente, essa ferramenta não provê funções específicas como o gerenciamento da topologia e a separação de planos. O sistema de gerenciamento MLN (*Manage Large Networks*) [57] define uma linguagem para construir e gerenciar uma rede virtual. MLN gerencia um ambiente de rede virtual que usa Xen ou outras tecnologias de virtualização como UML (*User Mode Linux*), mas não monitora funções importantes para se controlar e gerenciar o desempenho global do sistema. A ferramenta VNEXT, por outro lado, oferece a separação de planos para os roteadores virtuais e realiza um monitoramento dos enlaces e nós físicos e virtuais. Além disso, realiza a migração ao vivo do roteador virtual sem perda de

[1] `http://www.gtla.ufrj.br/vnext/`
[2] `http://www.gta.ufrj.br/horizon/`

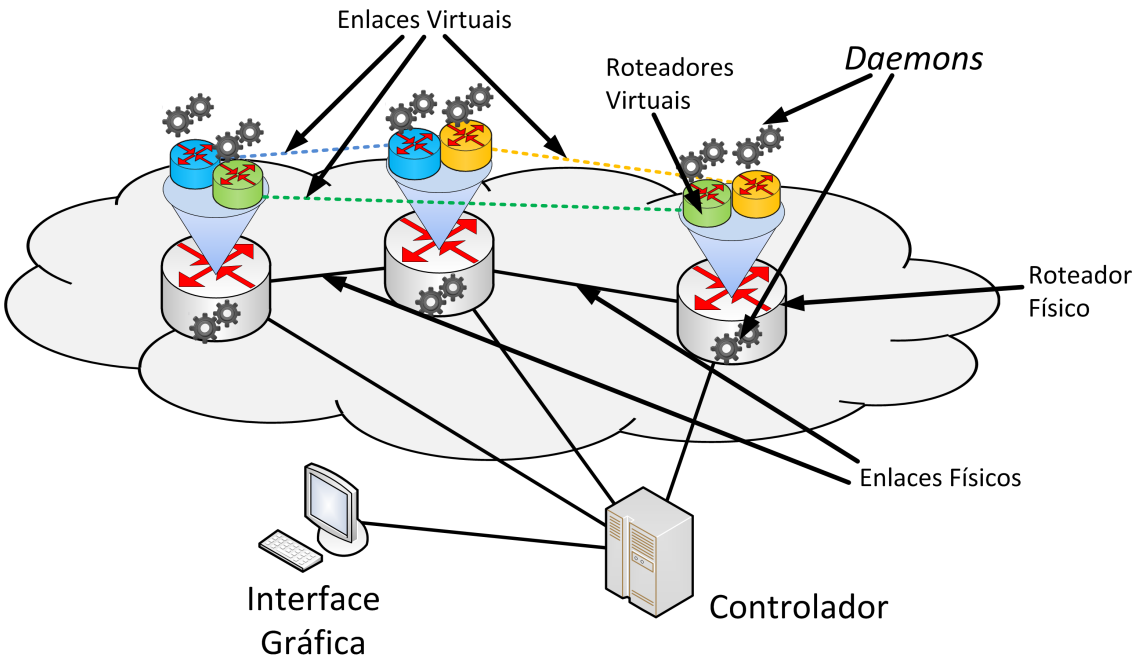


Figura 3.1: Arquitetura do VNEXT: controlador, *daemons* de monitoramento e atuação e interface gráfica de gerenciamento.

pacotes e oferece uma interface tridimensional para a virtualização da rede. Com a interface de usuário simples e amigável do VNEXT, é possível tomar decisões de alto nível, como redefinir a topologia da máquina virtual com o objetivo de economizar energia através do desligamento ou da colocação do roteador físico para dormir [58]. A contribuição dessa dissertação para a ferramenta consiste em três elementos: o mecanismo de separação de planos usado para a migração ao vivo de roteadores virtuais, a reformulação da arquitetura e no desenvolvimento de uma nova versão do controlador do VNEXT e, por fim, a interface para visualização da topologia das redes física e virtual.

3.1 Arquitetura

A arquitetura do VNEXT é composta por três componentes principais: um controlador da rede virtual, *daemons* para monitorar e atuar sobre roteadores físicos ou virtuais e uma interface gráfica para administradores da rede. O controlador exporta primitivas de gerenciamento e controle, que são disparadas pelo administrador da rede através da interface gráfica e interage com os *daemons* executados em cada roteador físico para implementar decisões do usuário. A Figura 3.1 apresenta a arquitetura funcional do VNEXT.

Os *daemons* de monitoramento e atuação são os principais componentes do sistema. Todo roteador físico e virtual na rede executa o *daemon*, cujas funcionalidades

básicas são: coletar as informações requisitadas pelo controlador, implementar as decisões do administrador nos roteadores e formatar a informação obtida de diferentes roteadores de acordo com padrões pré-estabelecidos. As informações coletadas consistem no uso de memória, no consumo de banda, no número de processadores e na porcentagem de uso do processador pelo sistema operacional e pelas aplicações do usuário. Todas essas informações são obtidas por medidas passivas e ativas realizadas por ferramentas desenvolvidas especificamente para o VNEXT, ou seja, ferramentas projetadas para ambientes de redes virtuais, e também por ferramentas existentes no Linux, como o `ifconfig`, o `top` e o `iptables`. Além de monitorar os roteadores, os *daemons* também são responsáveis por atuar nos roteadores físicos e virtuais, implementando as requisições do controlador como descrito na Section 3.2. Por exemplo, os *daemons* são responsáveis por migrar ou desligar roteadores virtuais mediante requisições do controlador. A comunicação entre os *daemons* é organizada hierarquicamente com objetivo de reduzir o tráfego de controle na rede. O controlador se comunica exclusivamente com os *daemons* dos roteadores físicos, que, por sua vez, se comunicam com os *daemons* dos roteadores virtuais em execução naquele roteador físico. Essa hierarquia melhora o isolamento entre as redes virtuais porque os roteadores virtuais não estão diretamente na rede de controle.

O controlador do VNEXT intermedia a comunicação entre administradores da rede e os *daemons* dos roteadores. O controlador recebe as requisições de monitoramento de pontos de medida do administrador da rede através da interface gráfica. Em seguida, o controlador envia comandos para os roteadores físicos que vão obter as medidas de desempenho solicitadas e as enviar para o controlador. O controlador pode ser executado em um computador específico ou em qualquer roteador físico da rede. O controlador proposto e desenvolvido em Java por Alves [59, 60]. Nesta implementação, os serviços são exportados como *Web Services*, usando a biblioteca Axis2 e o servidor Apache Tomcat6. Atualmente, o controlador é implementado em Python com mensagens em Notação de Objeto do JavaScript (JSON - *JavaScript Object Notation*) enviadas sobre o protocolo HTTP (*HyperText Transfer Protocol*) e o servidor em execução é o Apache. Com o aprimoramento da ferramenta VNEXT verificou-se a necessidade de se melhorar o desempenho e as funcionalidades do controlador. Esta dissertação apresenta uma nova versão deste controlador. A nova arquitetura proposta para o controlador é apresentada na Seção 3.4. Nesta versão, toda a comunicação entre o controlador e os *daemons* são realizadas por *sockets* TCP puros e mensagens no formato XML (*eXtensible Markup Language*). Todas as mensagens possuem um cabeçalho padrão para o XML, definido pela equipe do projeto VNEXT. O controlador também armazena o estado atual dos roteadores físicos e virtuais na rede, mantendo, por exemplo, uma lista com os roteadores físicos disponíveis e seus roteadores virtuais. O estado de atividade dos roteadores físicos

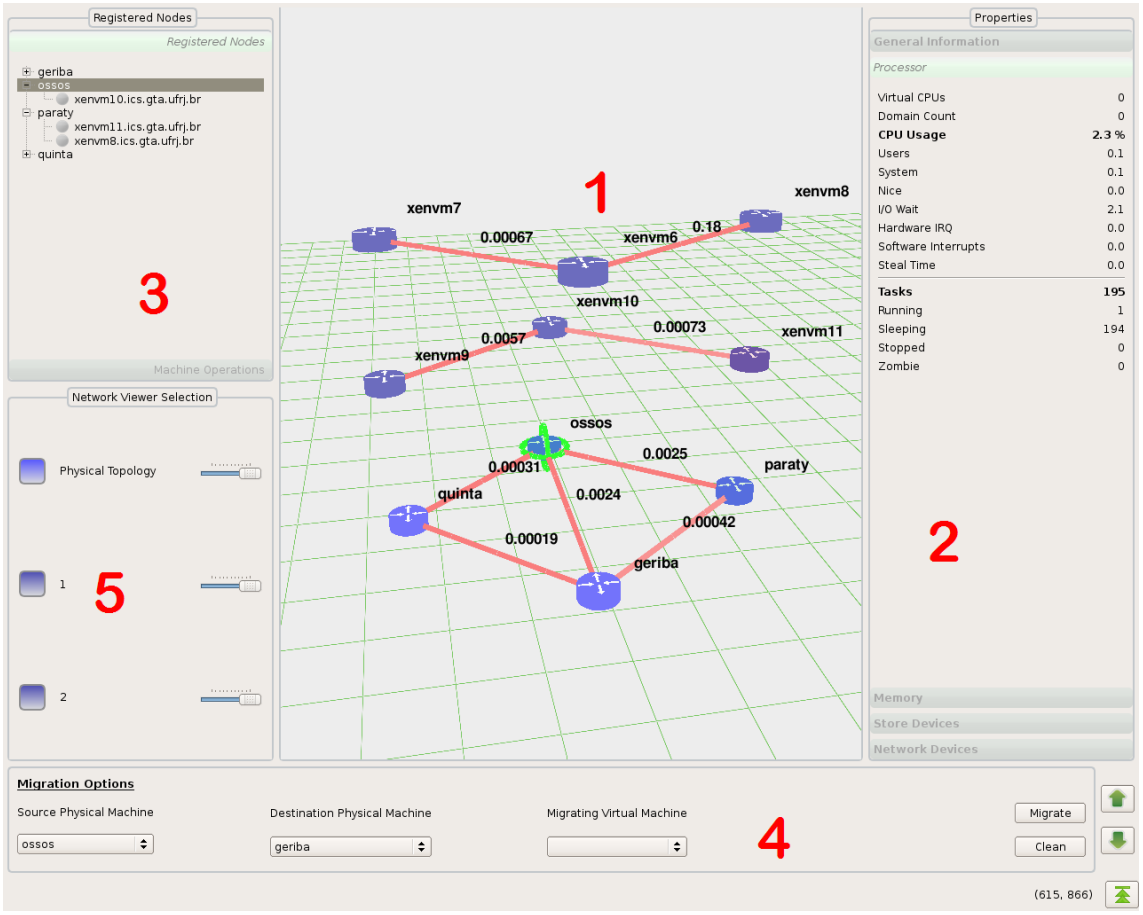


Figura 3.2: Tela inicial do VNEXT: (1) visualização da topologia da rede, (2) informações do roteador, (3) informações de disponibilidade do roteador, (4) mecanismos de migração e (5) preferências de personalização.

é monitorado por meio de mensagens enviadas periodicamente dos roteadores físicos para o controlador. Os estados dos roteadores virtuais, contudo, são mantidos através das requisições da biblioteca `Libvirt`, executada nos roteadores físicos para informar o estado do roteador virtual.

A interface gráfica (GUI - *Graphical User Interface*) fornece para o administrador um controle amigável e uma visualização das redes físicas e virtuais, como mostra a Figura 3.2. Ela foi desenvolvida em Python com as bibliotecas Qt e OpenGL. A GUI se comunica com o controlador através de chamadas os serviços no controlador, executados através de endereços específicos do protocolo HTTP para cada serviço. Na interface gráfica do VNEXT, as redes físicas e virtuais são estruturadas e exibidas como grafos em uma tela que fornece interações como rotação, translação e *zoom*. A interface gráfica inclui algoritmos para reorganização da topologia e para a descoberta do caminho mais curto entre dois nós baseado em duas métricas: número de saltos ou peso dos enlaces. Os pesos são, inicialmente, calculados de acordo com a latência do enlace, mas o administrador da rede pode definir valores fixos para esses enlaces. A GUI também mostra os roteadores físicos e virtuais que estão disponíveis no momento, as informações de monitoramento coletadas dos roteadores e as redes

que estão em execução sobre a infraestrutura naquele momento. Com respeito à atuação da rede, a GUI fornece telas para ligar e desligar os roteadores físicos e virtuais, e opções para reiniciar, suspender, retomar e migrar os roteadores virtuais. Usuários podem criar novas redes virtuais desenhando-as na janela de visualização da topologia. Após o desenho da topologia, as operações são processadas em lotes quando o usuário submete a nova rede para o controlador.

3.2 Principais Funcionalidades da ferramenta VNEXT

Esta seção apresenta resumidamente as principais funcionalidades do VNEXT.

Criação de roteadores virtuais

Uma rede virtual é composta por roteadores virtuais e enlaces virtuais conectando-os. No contexto da ferramenta VNEXT, um roteador virtual é uma máquina virtual Xen executando um protocolo de roteamento. Portanto, o administrador da rede define os requisitos do roteador virtual em termos de memória, CPUs virtuais e interfaces de rede. Os parâmetros da rede virtual são usados para garantir o isolamento do encaminhamento de pacotes quando o plano de dados está dentro do roteador virtual [61] ou compartilhando o encaminhamento do roteador físico [49]. O VNEXT cria um novo roteador a partir de uma imagem de disco de roteador pré-definida para evitar instalação do sistema operacional em tempo real. Além disso, após clonar a imagem do roteador, o VNEXT acessa o sistema de arquivos do novo roteador para instalar ferramentas adicionais de roteamento e configurar arquivos específicos para cada roteador virtual, como, por exemplo, as interfaces de rede e a configuração de arquivos do protocolo de roteamento. Essa funcionalidade fornece acesso ao administrador da rede e provê comunicação entre os novos roteadores virtuais porque define o endereço e as rotas iniciais. Uma vez criados os roteadores virtuais, eles podem ser iniciados a qualquer momento.

Criação de redes virtuais e controle individual da rede

A criação de redes virtuais sob demanda é fornecida aos operadores da rede. Após criar roteadores virtuais, o próximo passo é mapear os roteadores virtuais em roteadores físicos distintos e interligá-los através de enlaces virtuais. Portanto, criar uma rede virtual significa configurar as interconexões dos roteadores virtuais, definindo os endereços de IP da rede e escolhendo o protocolo de roteamento que será usado. O administrador da rede usa a interface gráfica para visualizar a topologia

física e, em seguida, criar uma nova rede virtual. O administrador define os requisitos dos roteadores e enlaces virtuais e a topologia da rede virtual desenhando os roteadores virtuais e os enlaces virtuais. A nova rede virtual é submetida como operação única no controlador. Essa operação pode ser executada em série ou em paralelo, para criar cada um dos roteadores virtuais. Após criar toda a rede virtual, ela pode ser iniciada imediatamente, ou seja, os roteadores da rede virtual podem ser ligados. O VNEXT pode também desligar ou remover redes virtuais para liberar recursos físicos.

Monitoramento de roteadores e enlaces

O monitoramento dos elementos da rede permite que o administrador da rede antecipe comportamentos futuros da rede e avalie a utilização dos recursos da rede [62]. A ferramenta desenvolvida pelo GTA fornece informações em tempo real relacionadas aos roteadores virtuais e seus enlaces. A principal janela da GUI exibe as topologias das redes físicas e virtuais, apresentando a latência dos enlaces em milissegundos. Quando o usuário seleciona qualquer roteador, as informações sobre o roteador selecionado são coletadas pelos *daemons* de monitoramento e exibidas no painel lateral. Entre as informações coletadas, destaca-se a latência atual das ligações físicas, a taxa e o número de pacotes transmitidos e recebidos, bem como o uso de recursos, como memória e CPU. Métricas de disco não foram consideradas por serem menos relevantes em roteadores.

Reorganização da topologia virtual

A virtualização da rede permite migrar um roteador virtual em funcionamento de um roteador físico para outro sem que seja necessário desligá-lo ou desativá-lo. Isso é possível apenas se todos os roteadores físicos executarem a mesma plataforma de virtualização. A vantagem da migração sobre uma simples cópia de roteador virtual é a manutenção do serviço de roteamento sem interrupções. O VNEXT implementa a migração com separação planos proposta por Pisa *et la.* [47], que migra um roteador virtual sem perda de pacotes durante o processo. Isso é possível porque o roteador não é desligado por completo e o plano de encaminhamento do roteador virtual é mantido funcional durante todo o processo de migração, sendo executado no roteador físico. Contudo, o VNEXT também implementa o padrão de migração Xen, que, por outro lado, pode perder pacotes.

Desligamento e religamento de roteadores físicos

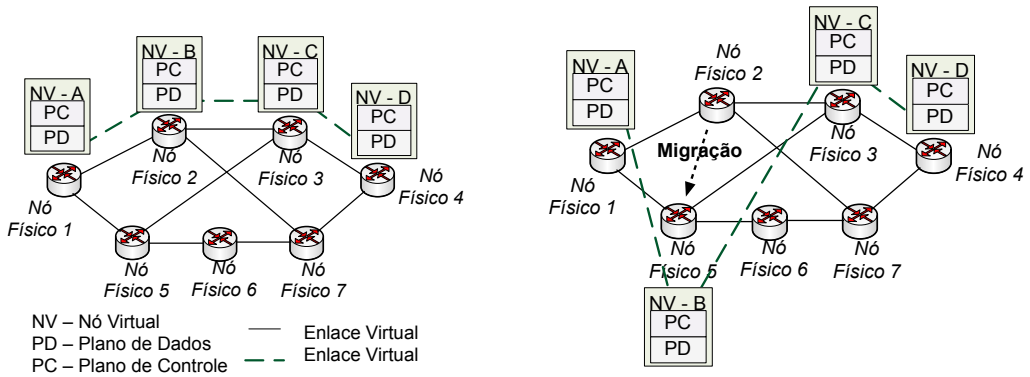
As propostas presentes na literatura possuem as funções de dormir e desligar os roteadores da rede para economizar energia [58, 63]. O VNEXT possui mecanismos para ligar e desligar roteadores físicos com o intuito de executar manutenção preventiva, recuperar falhas ou economizar energia em caso de baixa carga de rede. Quando o usuário faz uma requisição para que o sistema seja desligado, todos os roteadores virtuais são migrados antes para outro roteador físico. Na GUI, o administrador da rede escolhe o roteador físico para ser desligado e o roteador físico que irá receber os roteadores virtuais que precisam ser migrados. O processo de desligamento é realizado através de um comando para o *daemon* de atuação, que desliga o roteador físico. Por outro lado, para ligar remotamente um roteador físico, o VNEXT usa o mecanismo Wake-on-Lan, do padrão Ethernet.

3.3 Migração de Máquinas Virtuais

A migração da rede virtual é uma função para remapear a topologia da rede virtual sobre a topologia física. Tal remapeamento pode significar mudanças na topologia da rede virtual. O objetivo principal é mover enlaces e nós virtuais minimizando os impactos nas redes virtuais, como a necessidade de reconfigurar os roteadores virtuais, problemas na topologia virtual em nível de IP ou aumento no tempo de convergência dos algoritmos de roteamento das redes virtuais devido a perdas de mensagens causadas pelo processo de migração [17].

Há duas abordagens básicas para virtualizar um elemento de rede [23]. A primeira consiste em instalar uma plataforma de virtualização e fazer com que as máquinas virtuais ajam como roteadores virtuais, como é obtido pela plataforma Xen. Na segunda abordagem, utiliza-se um software de virtualização da rede para possibilitar que diversos administradores definam as regras de encaminhamento na mesma rede, como é feito no OpenFlow [45] em conjunto com o Flowvisor [64]. Nesta seção, aborda-se o funcionamento da migração de elementos virtuais considerando o modelo da virtualização do elemento de rede completo, realizada com a plataforma de virtualização do Xen.

Assume-se a existência de uma entidade que decide quando e para onde migrar nós e caminhos na rede. Essa entidade é ciente da topologia da rede física e das topologias de todas as redes virtuais e de suas cargas a cada momento, sendo, portanto, capaz de definir uma nova topologia da rede virtual que será migrada. A especificação e o desenvolvimento desta entidade complexa está fora do escopo desta dissertação, pois envolve aspectos de desempenho, centralizada ou distribuída, e de segurança. Uma entidade que tem poder sobre toda a rede ao se comunicar com os



(a) Visão da rede antes da migração do Nó B. (b) Visão da rede após da migração do Nó B.

Figura 3.3: Migração de rede sem separação de planos de controle e dados.

nós físicos e virtuais deve ter comunicações completamente seguras e não pode ser influenciada por nós de redes maliciosas que querem desviar recursos de uma rede para outra.

Nessa seção, apresenta-se o processo de migração padrão do Xen, proposto para servidores virtuais na Seção 3.3.1 e mostra-se porque esse processo não é adequado para a migração de roteadores virtuais. O procedimento proposto nessa dissertação e publicado por Pisa *et al.* [47] para roteadores virtuais é apresentado na Seção 3.3.2. A Seção 3.3.3 discute os resultados obtidos por Pisa *et al.* [47], que comparam os dois processos de migração.

3.3.1 Migração de Máquinas Virtuais no Xen

A forma de virtualização de redes na plataforma Xen pode ser vista como máquinas virtuais atuando como roteadores virtuais. Nessa abordagem, uma rede virtual é composta de nós virtuais, cada um contendo ambos os planos, de controle e de dados, e os enlaces virtuais, como mostra a Figura 3.3(a). Para manter a topologia da rede virtual, quando se migra um nó, deve-se encontrar um novo nó físico que também seja vizinho dos nós físicos que executam sobre os vizinhos do nó virtual a ser migrado. Em seguida, todo o ambiente virtual é migrado para o novo nó físico. Por exemplo, na Figura 3.3(b), o nó virtual B é migrado do Nó Físico 2 para o Nó Físico 5, porque essa mudança não muda a topologia da rede virtual.

Uma maneira de implementar essa abordagem de virtualização é pelo uso do Xen, uma plataforma de virtualização criada para um computadores padrão de mercado [65]. Na arquitetura Xen, um monitor da máquina virtual (VMM - *Virtual Machine Monitor*), também conhecido como hipervisor, é colocado sobre o hardware e fornece uma interface de processador virtual com a arquitetura x86 ou amd64 para a instanciação de máquinas virtuais (MVs). Assim, cada MV possui seus próprios

recursos virtuais, como CPU, memória e disco e também seu próprio sistema operacional e aplicações. O VMM agenda o acesso da máquina virtual ao processador físico, controla a divisão da memória entre máquinas virtuais e gerencia as interrupções geradas pelas MVs. Assim, a função do VMM é garantir que uma máquina virtual não interfira nas outras máquinas virtuais. O Xen também apresenta uma máquina virtual privilegiada, chamada Domínio 0, que tem acesso total aos dispositivos físicos e é responsável por fornecer suporte confiável para tarefas de entrada e saída para as outras máquinas virtuais.

Quando se usa o Xen para criar redes virtuais, assume-se que cada MV trabalha como um roteador virtual. Assim, migrar uma MV significa migrar um roteador virtual. No entanto, no caso da virtualização de redes, a MV está executando um serviço em tempo real, que é o encaminhamento de pacotes. Logo, há necessidade de se reduzir o *downtime* da máquina durante a migração, que é o tempo em que a máquina virtual não estará disponível para receber e enviar mensagens. É importante também minimizar o tempo total da migração para garantir que é possível liberar rapidamente os recursos da máquina física original que foi migrada. Xen possui um mecanismo embutido para migrar máquinas virtuais [66]. Esse mecanismo é baseado em algumas suposições: a migração ocorre dentro de uma rede local e o disco da máquina virtual está compartilhado em toda a rede[3]. A ideia principal desse procedimento é que migrar uma máquina virtual é o mesmo que copiar a memória da máquina virtual para o novo local físico e reconfigurar as ligações de rede sem quebrar conexões.

A forma mais simples de migrar a memória da MV é suspender a MV, transferir todas as páginas de memória para o novo local físico e, em seguida, retomar a execução da MV. No entanto, para reduzir o *downtime*, esse procedimento é transformado para uma migração com pré-cópia, na qual a cópia da memória é realizada em duas fases [66]. A primeira fase, chamada de pré-cópia iterativa, transfere, a cada iteração, todas as páginas de memória escritas na rodada anterior para a nova máquina física. As *hot pages* são as páginas modificadas frequentemente. Essas são as únicas páginas transferidas durante o *downtime*. Consequentemente, o *downtime* é reduzido, porque o montante total de páginas transmitidas é menor, ao invés de toda a memória. Nesse processo de pré-cópia iterativa, durante a primeira rodada, todas as páginas de memória são transferidas da origem para o destino com uma taxa mínima especificada pelo administrador da rede. Em seguida, nos outros turnos, apenas as

[3]A restrição do disco compartilhado pode ser reduzida no cenário de utilização proposto, porque roteadores do mesmo fornecedor geralmente implementam o mesmo conjunto pequeno de aplicações [67] e, em geral, não alteram os programas no disco. Em seguida, pode-se assumir que o roteador físico de destino da migração também possui esse conjunto de programas e é capaz de carregá-los no sistema de arquivos da nova MV. Por isso, apenas a memória do roteador virtual e os arquivos de configuração devem ser migrados.

páginas de memória que foram escritas pelo sistema operacional serão transferidas. A taxa de transferência é atualizada em cada iteração de acordo com um mecanismo de adaptação baseado na taxa de escrita das páginas da memória. Em cada rodada, a taxa de sujeira é calculada como a razão entre o número de páginas escritas na última rodada e a duração da última rodada. A taxa máxima de transmissão da próxima rodada é, em seguida, obtida por meio da adição de um incremento constante de 50 Mb/s sobre a razão de páginas sujas encontrada. A pré-cópia termina se a taxa máxima da rodada é igual à taxa máxima especificada pelo administrador ou menos de 256 kB de páginas escritas precisam ser transferidas. A próxima fase é chamada de pare-e-copie (*stop-and-copy*). Nessa fase, a máquina virtual é suspensa e as páginas escritas, principalmente as *hot-pages*, são transferidas com a taxa máxima de transferência do enlace. Em seguida, o novo nó físico de destino confirma a recepção de toda a memória para o nó físico de origem e retoma a execução da MV. Após o término da cópia, o nó físico de destino envia uma mensagem de ARP-Reply para todos os vizinhos, avisando que o MAC da interface da nova máquina passa a responder pelo IP do roteador virtual. Essa mensagem ARP-Reply é o mecanismo de migração dos enlaces virtuais.

A migração padrão do Xen é inadequada para a rede virtual devido à elevada perda de pacote durante o tempo de inatividade da MV. Embora a migração da pré-cópia reduza o tempo de inatividade, as perdas de pacotes continuam sendo altas considerando que o roteador virtual possui enlaces da ordem um gigabit por segundo. Outro problema da migração padrão do Xen para o uso em roteadores virtuais é que ela assume uma migração em uma rede local devido ao disco compartilhado e ao mecanismo de migração de enlaces utilizado, o que não compreende os objetivos da migração de roteadores virtuais. Na verdade, não é possível assumir que os nós físicos, como os Nós 1, 2 e 5 da Figura 3.3, sempre pertencem à mesma rede local.

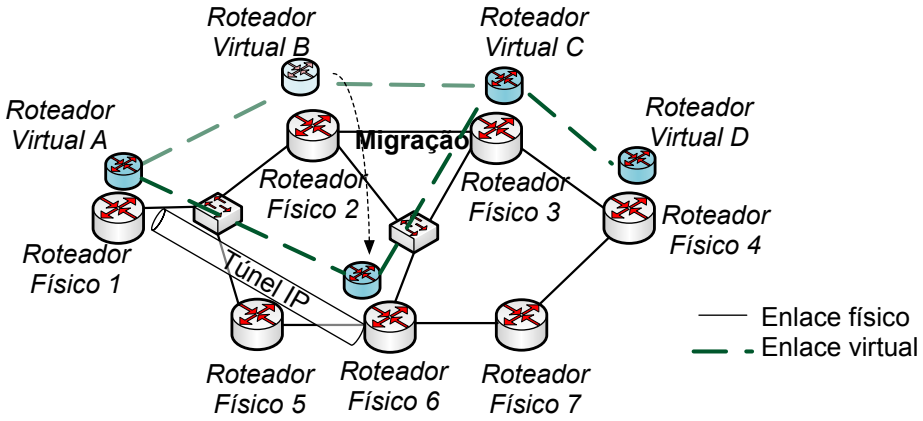
3.3.2 Migração de Roteadores Virtuais sem Perda de Pacotes

Wang et al. [67] propuseram uma nova abordagem de migração de máquinas virtuais mais apropriada para roteadores virtuais. A proposta consiste em separar os planos de dados e de controle para migrar o plano de controle sem perda de mensagens no plano de dados, o que é uma importante característica para a virtualização de roteadores virtuais. A proposta se baseia na plataforma de virtualização OpenVZ, que fornece múltiplos espaços de usuário virtuais sobre o mesmo sistema operacional. O Xen, contudo, apresenta uma plataforma de virtualização mais programável, porque cada roteador virtual pode ter seu próprio conjunto de software, incluindo sistema operacional e pilha de protocolos. Logo, Pisa *et al.* [47] propu-

seram o mesmo mecanismo de separação de planos para a migração de roteadores virtuais sem perdas de pacotes na plataforma Xen. Foi um protótipo que mantém o plano de controle na máquina virtual enquanto o plano de dados é implementado no Domínio 0. Cada roteador virtual possui sua própria tabela de encaminhamento no Domínio 0 e cada tabela é uma cópia da tabela de encaminhamento original, criada pelo software de roteamento executado na máquina virtual. Quando o Domínio 0 recebe uma mensagem de controle, ele verifica de qual rede essa mensagem pertence e encaminha a mensagem para a máquina virtual correspondente. Quando o Domínio 0 recebe uma mensagem do plano de dados, ele próprio a encaminha usando a tabela de encaminhamento correspondente àquela rede virtual.

O mecanismo de migração proposto funciona da seguinte maneira. Primeiro, a migração padrão do Xen é iniciada para migrar toda a máquina virtual. Durante esse processo, o encaminhamento dos pacotes continua funcionando no Domínio 0, com nenhuma interrupção ou perda de pacotes. Durante o período de inatividade da máquina, os pacotes de controle são armazenados na máquina física para posterior envio à máquina virtual na ordem em que foram recebidos. Quando toda a memória é copiada, o funcionamento da máquina virtual é retomado na máquina física de destino e as conexões de rede entre o Domínio 0 e a máquina virtual são criadas usando um módulo de conexão dinâmica das interfaces, que é responsável por mapear as interfaces virtuais nas interfaces físicas da nova máquina física. Depois disso, um túnel é criado da máquina de origem para a máquina de destino da migração com objetivo de transferir os pacotes de controle que estavam armazenados no Domínio 0 da máquina antiga e também os novos pacotes de controle recebidos nesse período. Com base nesses pacotes de controle recebidos, a máquina virtual é capaz de atualizar os planos de dados na máquina física de destino e de origem. Quando o plano de dados na máquina de destino está pronto para encaminhar os pacotes da rede virtual, a mensagem de ARP-Reply é enviada para atualizar os enlaces. Após toda a migração dos enlaces, o plano de dados na máquina de origem e o túnel criado são removidos para liberar recursos na máquina original.

O mecanismo de migração proposto garante que nenhum pacote do plano de dados será perdido durante a migração da máquina virtual, o que é uma característica importante para um roteador virtual. Além disso, também não há perdas de pacotes no plano de controle, uma vez que o mecanismo proposto insere apenas um atraso maior na entrega dos pacotes de controle da rede. O mecanismo proposto, contudo, é baseado na migração padrão do Xen, o que significa que os roteadores físicos ainda precisam estar na mesma rede local, com discos e enlaces compartilhados. O mapeamento dos enlaces virtuais sobre múltiplos enlaces físicos é ainda uma questão em aberto nessa proposta. Uma solução para esse mapeamento é a criação de túneis, como proposto por Mattos *et al.* [44], ou a instanciação de novos roteadores na rede.



1) Migração do plano de controle do roteador físico 2 para o 6, mantendo o plano de dados;
2) Remapeamento dinâmico das interfaces e criação do túnel entre os roteadores físicos 1 e 6;
3) Criação da nova tabela de encaminhamento no Domínio 0 do roteador físico 6;
4) Reconfiguração dos enlaces com o ARP-Reply;
5) Remoção do plano de dados no roteador físico 2.

Figura 3.4: Exemplo de migração de roteador no Xen quando um enlace virtual é mapeado para um caminho de múltiplos saltos na rede física.

Por exemplo, na Figura 3.4, migra-se o Nó Virtual B do Nó Físico 2 para o Nó Físico 6. O Nó Físico 6, no entanto, não é um vizinho de um salto do Nó Físico 1. Consequentemente, para completar a migração do enlace, precisa-se criar um túnel do Nó Físico 6 para o Nó Físico 1 com o objetivo de simular uma vizinhança de um salto. Outra solução é instanciar um novo roteador virtual no Nó Físico 5 para substituir o túnel. Essa solução, no entanto, modifica a topologia virtual e, com isso, influência no funcionamento do protocolo de roteamento. Portanto, a solução de instanciar um novo roteador não é desejável.

3.3.3 Avaliação dos Mecanismos de Migração

Para comparar ambas as abordagens de migração, foi desenvolvido um cenário de teste, descrito na Figura 3.5. A avaliação feita compara a migração padrão do Xen, que não possui separação de planos, com a implementação apresentada em Pisa *et al.* [47]. No cenário, existem dois roteadores físicos, Nó Físico A e Nó Físico B, que executam o Xen 4.0. O experimento consiste na migração de um roteador virtual do Nó Físico A para o Nó Físico B, enquanto esse roteador virtual encaminha um tráfego UDP da máquina Cliente para a máquina Servidor. Para não prejudicar o tráfego de dados, optou-se por utilizar um enlace exclusivo para transmitir o tráfego de migração durante o processo. Na migração padrão do Xen, o tráfego de dados é encaminhado pelo roteador virtual. Na abordagem do Xen com separação de planos, o tráfego de dados é encaminhado pelo plano de dados compartilhado no Domínio 0 e o tráfego de controle é encaminhado pelo roteador virtual.

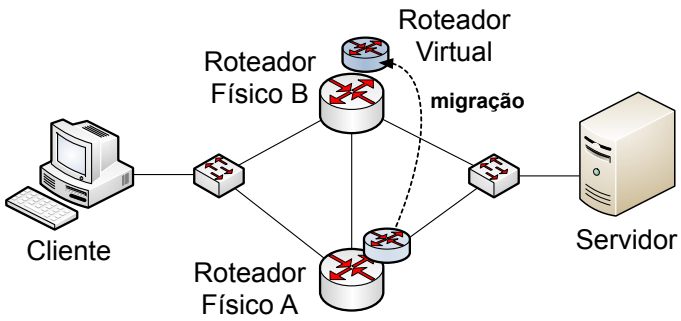
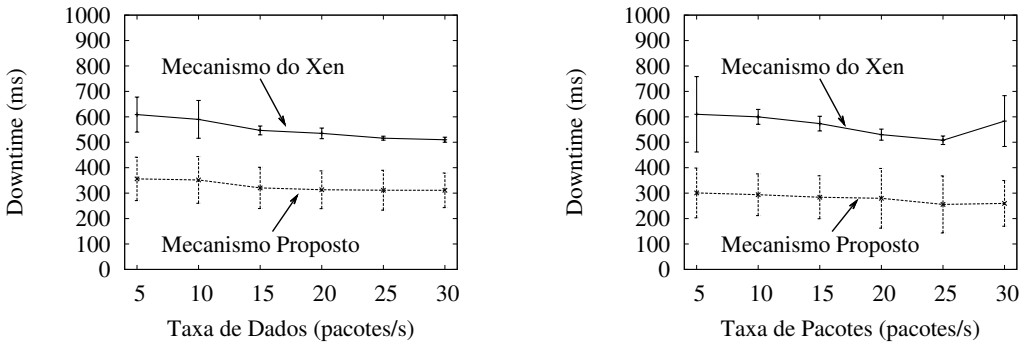


Figura 3.5: Cenário de teste para migração de um roteador virtualvirtual do Nó Físico A para o Nó Físico B, durante uma processo de tráfego UDP da máquina Cliente para a máquina Servidor.



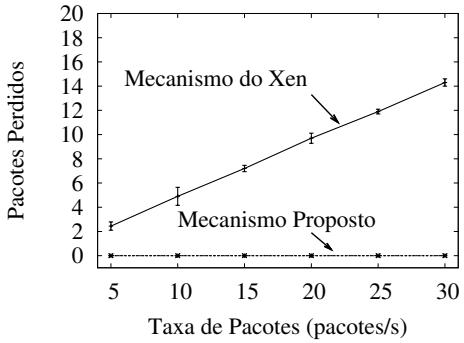
(a) Tamanho do pacote: 64 bytes.

(b) Tamanho do pacote: 1500 bytes.

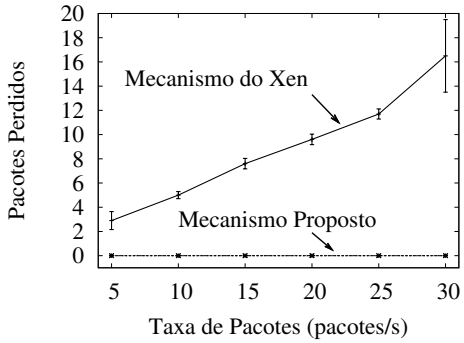
Figura 3.6: *Downtime* da migração em função da taxa de pacotes.

Nos experimentos realizados, avaliaram-se os atrasos e as perdas de pacotes relacionadas a migração, para diferentes taxas de pacotes de dados. Apresenta-se os resultados com pacotes de 64 bytes e de 1500 bytes, que correspondem, respectivamente, aos pacotes de menor tamanho do protocolo UDP e o tamanho de pacote máximo mais comum. A diferença no tamanho dos pacotes não representou grandes diferenças nos resultados obtidos.

A Figura 3.6 apresenta o tempo decorrido na fase de *downtime* do roteador virtual. Os resultados demonstram que o tempo de inatividade é aproximadamente constante com o crescimento da taxa de pacotes de dados para as configurações de migração avaliadas. Ao comparar o mecanismo de migração padrão do Xen com o mecanismo proposto, que possui separação de planos, o mecanismo proposto possui um tempo de inatividade, em média, de 200 milissegundos mais baixo. Essa diferença existe porque, na abordagem sem separação de planos, o roteador virtual precisa encaminhar todos os pacotes de dados e de controle, o que aumenta o número de *hot pages* na memória do roteador virtual e, consequentemente, aumenta o tempo de inatividade. Na abordagem com separação de planos, o tráfego de dados é

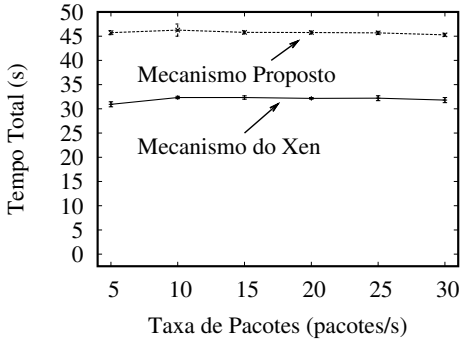


(a) Packet size: 64 bytes.

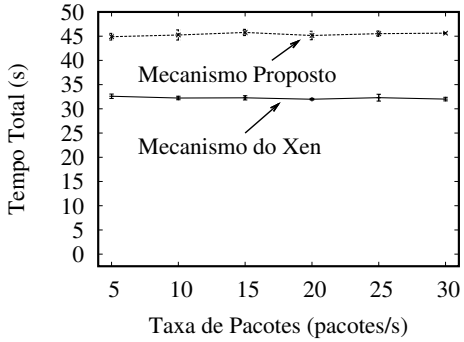


(b) Packet size: 1500 bytes.

Figura 3.7: Número de pacotes perdidos durante a fase de *downtime* em função da taxa de pacotes de dados.



(a) Tamanho do pacote: 64 bytes.



(b) Tamanho do pacote: 1500 bytes.

Figura 3.8: Tempo total de migração em função da taxa de pacotes de dados.

encaminhado pelo Domínio 0 e as páginas de memória do roteador virtual utilizadas durante a migração são oriundas somente do algoritmo de roteamento que executa no roteador virtual, o que reduz a inatividade do roteador virtual devido ao menor número de páginas a serem transmitidas.

Na Figura 3.7, analisa-se o número de pacotes de dados que foram perdidos durante a fase de *downtime*. Conforme esperado, no mecanismo de migração padrão do Xen, o número de pacotes perdidos aumenta linearmente com a taxa de pacotes, devido ao fato que o tempo de inatividade é constante e a taxa de pacotes aumenta linearmente. Por outro lado, a abordagem com separação de planos resolve esse problema, pois o plano de dados continua em funcionamento durante toda a migração, fazendo com que todos os pacotes de dados sejam encaminhados. A inatividade causada pela migração ocorre apenas no plano de controle, que é menos sensível a perdas de pacotes, visto que os algoritmos de roteamento possuem mecanismos de retransmissão ou entrega confiável de pacotes.

A Figura 3.8 apresenta o tempo total de migração, que consiste no tempo de-

corrido entre o disparo do mecanismo de migração e o momento em que os pacotes começam a passar pela máquina física de destino. A diferença entre o mecanismo de migração padrão do Xen e a abordagem com separação de planos é de aproximadamente 15 segundos. Esta variação ocorre, pois, após a migração do plano de controle, é necessário construir o plano de dados na máquina física de destino e mapear as interfaces virtuais do roteador virtual para as interfaces físicas apropriadas, o que não é feito no mecanismo padrão do Xen.

Com essa análise, conclui-se que a abordagem de separação dos planos de controle e de dados é uma característica fundamental para a solução do problema de perda de pacotes de dados durante a migração de roteadores virtuais. Além disso, devido a menor utilização do roteador virtual, que é utilizado apenas pelo plano de controle, o tempo de inatividade do plano de controle é menor, visto que menos páginas de memórias precisam ser transferidas enquanto o roteador virtual está inativo. No entanto, o mecanismo com separação de planos ainda não solutiona o problema da necessidade de mapeamento das interfaces de rede do roteador virtual nas interfaces de rede do roteador físico. Isso significa que a migração é limitada a roteadores físicos na mesma rede virtual e com enlaces redundantes.

3.4 Arquitetura do Controlador da Ferramenta

O controlador original da ferramenta VNEXT foi construído em Java, com a biblioteca Axis2 [59]. Esta implementação apresentava problemas, sendo que o principal deles era o mau funcionamento da biblioteca Libvirt para Java. Portanto, reformulou-se o controlador para uma implementação em Python e utilizando o protocolo HTTP nativo para a comunicação. Nesta nova versão do controlador foi desenvolvido um módulo de autenticação dos usuários, inexistente no controlador antigo. Após o usuário estar autenticado, o sistema gera uma chave de sessão e a envia para o usuário. Essa chave de sessão deve ser incluída em todas as requisições realizadas para garantir a identidade do usuário que utiliza os serviços do controlador da rede. Esse procedimento é o mesmo usado em sites da Web e em sistemas tradicionais, garantindo maior segurança.

Após receber as requisições, o controlador identifica o serviço invocado e procede com a validação dos parâmetros recebidos. Em seguida, o controlador aciona o gerenciador apropriado de acordo com o serviço em processamento. Como ilustra a Figura 3.9, existem quatro gerenciadores implementados no controlador: Gerenciador de Nós, Gerenciador de Topologia, Gerenciador de Roteador Físico e Gerenciador de Roteador Virtual. O Gerenciador de Nós é responsável por identificar os nós da rede que estão ativos, mantendo o registro deles. Através desse gerenciador, é possível que um administrador monitore os nós registrados na rede em tempo real, veja

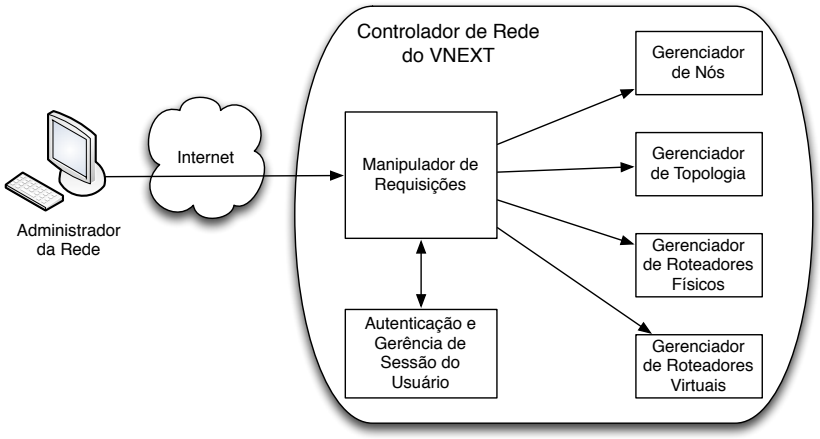


Figura 3.9: Arquitetura do controlador implementado para o VNEXT.

a listagem de todos os nós já registrados, bem como realizar um registro manual ou mesmo retirar um nó da rede. Já o Gerenciador da Topologia mantém o registro dos roteadores físicos presentes na rede. Esse cálculo da topologia virtual é realizado periodicamente, pois deve-se coletar em cada roteador físico da rede os seus nós virtuais e cada vizinho desses nós. Quando o usuário requisita a topologia da rede, o gerenciador envia o resultado do último cálculo. A topologia virtual é calculada localmente no controlador de acordo com o arquivo de configuração da rede e dos roteadores que fazem parte de cada rede.

A tarefa do Gerenciador de Roteadores Físicos é obter informações dos nós físicos, como estatísticas de uso do processador, consumo de memória e tráfego de rede. Esse gerenciador ainda é capaz de ligar uma máquina física desligada, através do serviço Wake-on-Lan, do padrão Ethernet, e de desligar uma máquina ativa para economizar energia [58]. Quando um roteador físico é desligado, mas ele ainda possui roteadores virtuais ativos, o gerenciador é responsável por invocar o Gerenciador de Roteadores Virtuais para migrar cada um dos roteadores virtuais ainda em execução antes do desligamento da máquina. O Gerenciador de Roteadores Virtuais, além de ser responsável pelo processo de migração, realiza as tarefas de criação e instanciação de novos roteadores virtuais, bem como a coleta de estatísticas, o reinício, o desligamento, a suspensão e a retomada de roteadores virtuais em execução.

Parte II

Criptografia Homomórfica

Capítulo 4

Encriptação Homomórfica

O conceito de Encriptação Homomórfica, inicialmente proposto por Rivest, Adleman e Dertouzos [18] com o nome de Homomorfismo Secreto (*Private Homomorphism*), surgiu pouco tempo depois da proposta do algoritmo RSA, por Rivest, Shamir e Adleman [68]. Um esquema de encriptação é dito homomórfico se ele permitir operações sobre os dados encriptados sem a necessidade de decriptá-los. A partir dessa definição, o algoritmo RSA padrão é homomórfico para operações de multiplicação, pois é possível multiplicar textos cifrados obtidos a partir do RSA padrão para obter o texto cifrado do produto. Depois da multiplicação, basta decriptar o resultado para recuperar o resultado desejado. A validação dessa propriedade segue do fato do RSA ser baseado na exponenciação. De fato, dado uma chave pública $K_{pub} = (N, e)$, pode-se obter o texto cifrado Ψ através da expressão $\{\Psi \leftarrow \pi^e \bmod N\}$. Logo, k textos cifrados podem ser multiplicados como

$$\prod_{i}^{k} \Psi_i = (\prod_{i}^{k} \pi_i)^e \bmod N, \tag{4.1}$$

desde que o valor total não ultrapasse N. Após o produtório, recupera-se o resultado usando a chave privada K_{priv} para decriptar $\prod_i^k \Psi_i$. Contudo, apenas as operações de multiplicações são executadas homomorficamente com o RSA. Isso ocorre devido ao algoritmo se basear em exponenciação, que não possui propriedades diretas para operações de adição. Portanto, o RSA padrão é considerado como um algoritmo parcialmente homomórfico. Para que um esquema de encriptação seja considerado totalmente homomórfico há necessidade de que as operações de adição e multiplicação sejam homomórficas [69]. Desde a criação do conceito de homomorfismo, estima-se que essa técnica pode revolucionar as áreas de segurança e privacidade [70], por permitir processamento seguro em ambientes não confiáveis, como computação em nuvens, grades distribuídas e *testbeds* compartilhados. Esse processamento seguro possui aplicações em diversas áreas, como o processamento de extratos bancários ou

operações ações, votações eletrônicas e prontuários médicos.

O esquema de encriptação totalmente homomórfica de Craig Gentry [19] foi o primeiro esquema a demonstrar a viabilidade do conceito mantendo a segurança dos dados. No entanto, o método é complexo e possui alto custo computacional. A partir do trabalho de Craig Gentry, Marten Van Dijk *et al.* [1] propuseram um esquema de encriptação totalmente homomórfica mais simples. O esquema é baseado na aritmética modular de números inteiros[1] e preserva a segurança do esquema proposto por Craig Gentry.

4.1 Príncipios de Segurança e Criptografia

Nesta seção, são explicados alguns conceitos de segurança de esquemas criptográficos que serão usados ao longo da dissertação. Esses conceitos envolvem os princípios de provas de segurança dos esquemas e os modelos de desafios que os adversários realizam contra os esquemas criptográficos.

Criptografia de chaves simétricas

Os algoritmos de chaves simétricas utilizam a mesma chave criptográfica tanto para encriptação da mensagem quanto para decriptação do texto cifrado. Esses algoritmos se dividem em dois grupos de cifragem: por fluxo e por bloco. A cifragem por fluxo consiste em criptografar os dígitos ou os bits de uma mensagem um por vez. Em geral, eles são usados para encriptação da informação conforme ela é recebida ou gerada. Já as cifras por blocos consideram um número maior de bits por vez. Em geral, as cifras utilizam blocos de 128 ou 256 bits. Por tratar diversos bits por vez, essa cifragem apresenta melhor desempenho e é largamente utilizada. O padrão de segurança americano, o algoritmo *Advanced Encryption Standard* (AES) [71], é um algoritmo de cifragem por blocos.

Na criptografia simétrica, as chaves podem ser idênticas ou diferenciadas por uma simples transformação. Essas chaves, na prática, representam um segredo compartilhado entre duas ou mais pessoas que pode ser usado para manter a informação privada. Esse requisito de que ambas as partes precisam ter acesso à chave secreta é um dos maiores inconvenientes da criptografia de chaves simétricas em comparação com os esquemas de criptografia de chaves assimétricas. Nessa abordagem, as partes precisam estabelecer, de forma segura, a chave secreta.

[1]Deste ponto em diante, nessa dissertação, esse esquema é chamado de DGHV, que são as iniciais dos autores.

Criptografia de chaves assimétricas

A criptografia de chaves assimétricas ou criptografia de chaves públicas refere-se ao sistema criptográfico que requer duas chaves separadas, sendo uma secreta, denominada chave privada, e a outra, pública. Embora diferentes, as duas chaves são matematicamente interligadas. No caso de um sistema de encriptação, uma chave é usada para criptografar a mensagem e a outra para decriptar o texto cifrado. Nenhuma das duas chaves pode realizar as duas funções sem a outra chave. Enquanto que a chave pública é distribuída livremente, a chave privada não deve ser revelada para ninguém. A chave privada é uma chave pessoal e, por isso, o sigilo só depende da pessoa que a possui.

A criptografia de chaves assimétricas usa algoritmos como o RSA (Rivest, Shamir e Adleman) [18]. Esses algoritmos são baseados em relações matemáticas que supostamente não possuem soluções eficientes, como os problemas de fatoração de números inteiros e de logaritmos discretos. Embora seja computacionalmente fácil para o destinatário gerar ambas as chaves pública e privada, seja fácil para o emissor criptografar a mensagem usando a chave pública recebida e igualmente fácil para o destinatário descriptografar a mensagem com sua chave privada, é extremamente difícil ou praticamente impossível para qualquer outro usuário que não o dono da chave privada derivar a chave privada a partir da chave pública. Esse é o motivo principal pelo qual os algoritmos de chaves públicas não requerem uma troca inicial segura das chaves secretas entre o emissor e o destinatário.

Indistinguibilidade e segurança semântica

Em criptografia, um esquema é semanticamente seguro se, dado um texto cifrado de uma mensagem qualquer e o tamanho dessa mensagem, não houver um algoritmo probabilístico em tempo polinomial (*Probabilistic, Polynomial-Time Algorithm* - PPTA) que determine qualquer parte da informação da mensagem com probabilidade significantemente maior do que uma escolha aleatória [72]. Em outras palavras, o conhecimento do texto cifrado e do tamanho do espaço de mensagens sobre uma mensagem desconhecida não revela nenhuma informação sobre a mensagem que possa ser extraída facilmente do texto cifrado. Esse conceito tem complexidade computacional semelhante ao conceito de segredo perfeito (*perfect secrecy*) de Shannon [73]. No entanto, enquanto que o conceito de segredo perfeito significa que o texto cifrado não revela nenhuma informação sobre a mensagem, a segurança semântica implica que qualquer informação revelada não pode ser facilmente extraída.

A indistinguibilidade dos textos cifrados [74] é uma propriedade de muitos esquemas de criptografia. Intuitivamente, se um sistema de criptografia tem essa

propriedade, um atacante não é capaz de distinguir entre dois textos cifrados com base na mensagem que cada texto cifrado guarda. Essa propriedade contra o ataque do texto claro escolhido (*Chosen Plaintext Attack* - CPA) é considerada como requisito básico de segurança para muitos esquemas de criptografia de chaves públicas. Alguns esquemas também fornecem indistinguibilidade contra ataques de texto cifrado escolhido (*Chosen Ciphertext Attack* - CCA) e contra ataques adaptativos de texto cifrado escolhido (*Adaptive Chosen Ciphertext Attack* - CCA2). A prova de que um esquema criptográfico é semanticamente seguro consiste no fato de que os textos cifrados são indistinguíveis frente ao CPA.

Um esquema é considerado seguro em termos da indistinguibilidade se nenhum adversário, dada a encriptação de uma mensagem escolhida aleatoriamente entre duas possibilidades, for capaz de identificar a mensagem criptografada com probabilidade significantemente melhor do que a escolha aleatória (1/2). Se qualquer adversário puder ter sucesso em distinguir o texto cifrado escolhido com uma probabilidade significantemente maior do que 1/2, então o esquema não é considerado seguro em termos de distinguibilidade. Essa definição engloba a noção de que em um esquema seguro, um adversário não deve conseguir obter nenhuma informação a partir do texto cifrado. Portanto, o adversário não deve ser capaz de fazer nada melhor do que uma escolha aleatória.

A segurança em termos da indistinguibilidade tem muitas definições e depende das premissas feitas sobre as capacidades dos atacantes, o que determina se o esquema possui segurança contra CPA, CCA ou CCA2. A prova de segurança é geralmente apresentada como um jogo ou desafio, em que o esquema de criptografia é considerado seguro se não existir adversário capaz de vencer o jogo com probabilidade significantemente maior do que outro adversário que chuta aleatoriamente o resultado. A seguir, são definidos os jogos aplicados a cada tipo de premissa de atacante.

O Ataque do Texto Claro Escolhido

A indistinguibilidade contra o ataque de texto claro escolhido é definido como um jogo entre um adversário e um desafiante. O adversário é definido como um algoritmo que gera uma saída em tempo polinomial. Os passos do jogo são definidos a seguir:

1. o desafiante gera um par de chaves (K_{priv}, K_{pub}) baseado em algum parâmetro de segurança λ, que representa, por exemplo, o tamanho em bits da chave privada. Em seguida, o desafiante publica a chave pública para o adversário e retém a chave privada;

2. o adversário pode realizar qualquer quantidade de encriptações ou outras operações;

3. em algum momento, o adversário envia para o desafiante duas mensagens π_0 e π_1;

4. o desafiante escolhe um bit $b \in \{0, 1\}$ de forma uniformemente aleatória e envia o texto cifrado $\Psi = ENC(K_{pub}, \pi_b)$ para o adversário;

5. o adversário pode realizar qualquer quantidade de encriptações ou outras operações, novamente;

6. no final, o adversário tenta acertar o valor escolhido de b pelo desafiante.

Com a repetição desse procedimento diversas vezes, calcula-se a probabilidade de sucesso do adversário, ou seja, a probabilidade do adversário acertar o valor de b. Assim, define-se a vantagem do adversário, em função de λ, sobre o esquema criptográfico pela expressão

$$\epsilon(\lambda) = P(SucessoAdversário) - \frac{1}{2}. \tag{4.2}$$

Se o valor de $\epsilon(\lambda)$ não for negligenciável, o esquema criptográfico não é seguro para esse valor do parâmetro de segurança (λ).

O Ataque do Texto Cifrado Escolhido

Nesse caso, além das capacidades do adversário que realiza o CPA, o adversário tem acesso a um oráculo de decriptação, ou seja, ele é capaz de realizar decriptações de textos cifrados que ele gere para obter a mensagem resultante. No cenário padrão, o usuário só tem acesso ao oráculo de decriptação antes de enviar as duas mensagens para encriptação do desafio. O esquema é considerado seguro nas mesmas condições descritas para o modelo CPA.

O Ataque Adaptativo de Texto Cifrado Escolhido

Para o caso do ataque adaptativo de texto cifrado escolhido, o adversário é capaz de usar o oráculo de decriptação antes e depois de enviar as mensagens para o desafio. No entanto, uma restrição é feita para que o resultado não seja trivial. Após o envio do desafio para o adversário, este não pode reenviar o mesmo texto cifrado para o oráculo de decriptação. Se pudesse, o adversário sempre seria capaz de acertar o desafio. O esquema é considerado seguro nas mesmas condições do modelo CPA e do modelo CCA.

4.2 Principais Conceitos da Encriptação Homomórfica

Esta seção apresenta os principais conceitos de encriptação homomórfica que serão usados ao longo deste e do próximo capítulo, como, dentre outros, a corretude, a privacidade do circuito de computação, a compactação dos textos cifrados e a segurança circular.

4.2.1 Circuito de Computação Φ

Um circuito Φ corresponde a um conjunto de operações aplicadas sobre o texto cifrado. Para o caso de esquemas de encriptação capazes de encriptar apenas bits, o circuito pode ser visto como um circuito de portas lógicas. Por outro lado, quando o esquema encripta números inteiros de qualquer tamanho, o circuito é semelhante a uma função algébrica. A complexidade de um circuito Φ é dada pela complexidade de suas operações e pela sua profundidade d. A profundidade de um circuito é a quantidade de operações a serem realizadas sequencialmente no circuito desde as entradas até a obtenção da saída do circuito.

4.2.2 Corretude

Um esquema de encriptação padrão possui três algoritmos básicos: *Geração de Chaves* (*KeyGen*), *Encriptação* (*ENC*), and *Decriptação* (*DEC*). Em um esquema homomórfico, adiciona-se um quarto algoritmo, chamado de *Avaliação* (*EVAL*), que objetiva aplicar a operação desejada sobre os textos cifrados e obter o texto cifrado do resultado. Assim, o esquema homomórfico é denotado por $\mathfrak{E} = (KeyGen, ENC, DEC, EVAL)$. O algoritmo de *Avaliação* é denotado por

$$\Psi' = EVAL(K_{pub}, \Phi, \vec{\Psi}) \tag{4.3}$$

e executa o circuito Φ para o vetor $\vec{\Psi}$, com o auxilio da chave pública K_{pub}. Embora a corretude de um esquema de encriptação homomórfica seja basicamente a mesma de qualquer esquema de encriptação padrão, a prova de sua corretude requer a decriptação do texto cifrado produzido pelo algoritmo de *Avaliação*. Assim, o esquema de encriptação homomórfica é correto para qualquer operação Φ se a equação

$$DEC(K_{priv}, EVAL(K_{pub}, \Phi, \vec{\Psi})) = \Phi(\pi_1, \pi_2, \cdots, \pi_n) \tag{4.4}$$

é válida [1], para qualquer par de chaves (K_{pub}, K_{priv}) gerado pelo algoritmo de *Geração de Chaves*, qualquer número n de mensagens $\pi_1, \pi_2, \cdots, \pi_n$ e seus corres-

pondentes textos cifrados $\vec{\Psi} =< \Psi_1, \Psi_2, \cdots, \Psi_n >$, gerados a partir do algoritmo de *Encriptação*. Portanto, define-se o conjunto S_Φ contendo todos os circuitos Φ corretos para o esquema $\mathfrak{E}$.

4.2.3 Encriptação Totalmente Homomórfica

Quando um esquema homomórfico é capaz de processar corretamente cada um dos circuitos Φ de um conjunto S_Φ de circuitos, diz-se que ele é correto para S_Φ. Assim, um esquema totalmente homomórfico é aquele em que o conjunto S_Φ compreende todos os circuitos algébricos ou booleanos.

4.2.4 Privacidade do Circuito de Computação

Um esquema de encriptação possui *privacidade do circuito* se não for possível descobrir as operações realizadas até obter o texto cifrado corrente. Na criptografia tradicional, essa propriedade não é relevante, visto que o texto cifrado foi gerado apenas pela operação de encriptação. No entanto, na encriptação homomórfica, diferentes operações podem ter sido aplicadas para se obter o texto cifrado corrente. Portanto, é importante proteger as operações aplicadas, que podem ser secretas. Assim, diz-se que um esquema de encriptação possui privacidade de circuito se não for possível distinguir entre a encriptação do resultado do circuito e a saída do algoritmo de *Avaliação* aplicado ao mesmo circuito, ou seja,

$$ENC(K_{pub}, \Phi(\pi_1, \pi_2, \cdots, \pi_n)) \approx EVAL(K_{pub}, \Phi, \vec{\Psi}). \tag{4.5}$$

Assim, não é possível saber se o texto cifrado foi simplesmente encriptado ou se foi processado homomorficamente.

4.2.5 Compactação dos Textos Cifrados

A compactação dos textos cifrados é uma relação entre o tamanho do texto cifrado gerado pelo algoritmo de *Encriptação* e o gerado pelo algoritmo de *Avaliação*. Assim, independentemente da profundidade do circuito Φ, o texto cifrado gerado após o processamento homomórfico é polinomial em relação ao parâmetro de segurança λ. Esse parâmetro é um valor que define todos os outros parâmetros do sistema e quanto maior é o valor de λ, maior é a segurança do esquema de encriptação. Dessa forma, um esquema compacto é aquele em que

$$log_2\ (EVAL(K_{pub}, \Phi, \vec{\Psi})) \approx log_2\ (ENC(K_{pub}, \pi_1)) \approx log_2\ (ENC(K_{pub}, \pi_2)), \tag{4.6}$$

em que a aplicação do logaritmo em base 2 tem o objetivo de calcular o tamanho em bits do texto cifrado gerado.

4.2.6 Circuito de Decriptação Aumentado

Considerando um esquema de encriptação homomórfica definido por $\mathfrak{E} = (KeyGen, ENC, DEC, EVAL)$, no qual o algoritmo de *Decriptação* depende apenas do parâmetro de segurança λ. Para um dado valor do parâmetro de segurança λ, o conjunto de circuitos de decriptação aumentado consiste em dois circuitos: o circuito de decriptação aumentado para adição ($DEC^{+}_{\mathfrak{E}}$) e o circuito de decriptação aumentado para multiplicação ($DEC^{\times}_{\mathfrak{E}}$). Ambos os circuitos aumentados recebem como entrada a chave privada (K_{priv}) e dois textos encriptados (Ψ_1 e Ψ_2). O processamento dos circuitos aumentados é dividido em duas etapas. Primeiro, decripta-se os textos encriptados e, após, realiza-se da operação desejada, que pode ser uma adição ou uma multiplicação.

4.2.7 Encriptação com Auto-inicialização

Dado um esquema $\mathfrak{E} = (KeyGen, ENC, DEC, EVAL)$ de encriptação homomórfica, para cada valor do parâmetro de segurança λ, assuma $S_{\Phi}(\lambda)$ como o conjunto de circuitos com os quais $\mathfrak{E}$ é correto. Caso $DEC^{+}_{\mathfrak{E}} \subseteq S_{\Phi}(\lambda)$ e $DEC^{\times}_{\mathfrak{E}} \subseteq S_{\Phi}(\lambda)$ seja válido para todo valor de λ, então diz-se que o esquema de encriptação totalmente homomórfico é auto-inicializável.

4.2.8 Encriptação Homomórfica em Níveis

O funcionamento do esquema de encriptação homomórfica em níveis é simples. Em cada nível do circuito Φ, os textos cifrados são re-encriptados usando a chave pública do nível seguinte, $K_{pub_{i+1}}$. Além disso, cada operação do circuito Φ é substituída pelo circuito de decriptação aumentado $DEC_{\mathfrak{E}}$ apropriado. Esse algoritmo que processa um nível do circuito Φ é chamado de *Re-encriptação*.

Dado que $\mathfrak{E}$ é auto-inicializável, pode-se definir um esquema $\mathfrak{E}^d$ homomórfico para todos os circuitos Φ de profundidade d. A segurança semântica do novo esquema $\mathfrak{E}^d$ é dada pela segurança semântica do esquema $\mathfrak{E}$ original. Qualquer ataque $\mathfrak{A}$ com vantagem ϵ contra $\mathfrak{E}^d$ pode ser convertido em um ataque com complexidade similar contra o esquema $\mathfrak{E}$ com vantagem de, no mínimo,

$$\frac{\epsilon}{\ell d}, \tag{4.7}$$

em que ℓ é o tamanho da chave privada no esquema $\mathfrak{E}$.

Esse novo esquema definido, $\mathfrak{E}^d$, é compacto e possui o mesmo algoritmo de *Decriptação* do esquema original $\mathfrak{E}$. Além disso, os textos cifrados possuem o mesmo tamanho dos gerados pelo esquema original $\mathfrak{E}$. No entanto, o par de chaves sofre modificações e consiste em $d+1$ pares de chaves do esquema $\mathfrak{E}$. Após a geração dos pares de chaves, calcula-se a encriptação de cada uma das chaves privadas K_{priv_i} pela chave pública do nível seguinte, $K_{pub_{i+1}}$.

4.2.9 Segurança Circular

Um esquema $\mathfrak{E}$ de encriptação homomórfica possui segurança circular [75] caso a encriptação da chave privada com a sua própria chave pública for seguro. A partir desse conceito, se um esquema possui a segurança circular, não é preciso ter $d+1$ pares de chaves para aplicar a encriptação em nível. Nesse caso, apenas um par de chaves é necessário, pois a própria chave pública é usada para encriptar a chave privada em todos os níveis de utilização da decriptação homomórfica.

4.3 Primeiras Propostas de Criptografia Homomórfica

A noção de esquemas de encriptação que permitissem operações com os dados encriptados foi inicialmente proposta por Rivest, Adleman e Dertouzos [68], que propuseram a função exponencial

$$\Psi \leftarrow \pi^e \bmod N \tag{4.8}$$

como homomorfismo aditivo e o RSA [18] como homomorfismo multiplicativo. No entanto, esses métodos de criptografia não proveem segurança contra ataques texto claro escolhido (Chosen Plaintext Attack - CPA), não sendo, portanto, semanticamente seguros. O esquema de encriptação El Gamal [76], derivado da função de exponenciação, é semanticamente seguro e homomórfico apenas para operações de multiplicação.

A principal característica que diferencia o esquema de encriptação de ElGamal [76] da função de exponenciação e do RSA é a presença de uma componente aleatória no cálculo. No esquema de ElGamal, a chave privada a é escolhida aleatoriamente no intervalo $(0, q-1)$, onde q é o tamanho do subgrupo de ordem q de $\mathbb{Z}_p^*$, normalmente definido como $p = 2q+1$ para a maior segurança possível. Para a definição da chave pública, calcula-se $A = g^a \bmod\ p$, em que g é um gerador de G, um subgrupo de ordem prima de $\mathbb{Z}_p^*$. A escolha de G deve ser feita com base nas premissas de decisão de Diffie-Hellman (DDH), as quais determinam que, dado g^a e

g^b para valores de a e b escolhidos aleatoriamente no intervalo $(0, q-1)$, o valor g^{ab} é computacionalmente indistinguível de g^c, com c também escolhido aleatoriamente. Assim, para que o problema DDH seja difícil, q tem que ser um número primo grande. Como a ordem de $\mathbb{Z}_p^*$ é $2q$, o subgrupo de ordem q é o maior subgrupo de ordem prima. No algoritmo de ElGamal, a encriptação de uma mensagem $\pi \in \mathbb{Z}_p^*$ é feita através da expressão

$$(\Psi_1, \Psi_2) = (g^k, A^k \times \pi), \tag{4.9}$$

em que k é um número aleatório no intervalo $(0, q-1)$. A decriptação, por sua vez, é realizada pela expressão

$$\pi = (\Psi_1^a)^{-1} \times \Psi_2 (\text{mod } p). \tag{4.10}$$

A partir dessa definição do esquema, pode-se definir a operação de multiplicação homomórfica. Para isso, dados dois textos cifrados, (Ψ_1, Ψ_2) e (Ψ_1', Ψ_2'), o resultado da multiplicação é dado por

$$(\Psi_1, \Psi_2) \times (\Psi_1', \Psi_2') = (\Psi_1 \times \Psi_1', \Psi_2 \times \Psi_2') = (g^{k+k'}, g^{a(k+k')} \times \pi \times \pi'). \tag{4.11}$$

Portanto, observa-se que o esquema de ElGamal é semanticamente seguro e parcialmente homomórfico para operações de multiplicações.

Proposto antes do esquema de ElGamal, o primeiro esquema de encriptação homomórfica semanticamente seguro é o trabalho de Goldwasser e Micali (GM) [72], que define a primeira noção robusta de segurança para encriptação. O esquema de Goldwasser e Micali baseia-se no problema intratável de determinar se um número x é um resíduo quadrático no conjunto $\mathbb{Z}_N$, em que $N = p.q$, com p e q primos grandes, ou seja, determinar se $x = y^2 \bmod N$ para algum y. Por outro lado, se tivermos a fatoração de N, esse problema é fácil de ser resolvido. Assim, calcular o resíduo quadrático em $\mathbb{Z}_p$ ou $\mathbb{Z}_q$ é trivial. Goldwasser e Micali se utilizaram dessa propriedade para definir o seu criptossistema, o qual possui a chave privada definida como (p, q) e a chave pública por (N, z), onde $z \in \mathbb{Z}_N | z^{\frac{p-1}{2}} \equiv 1 \bmod N$ e z não é um resíduo quadrático em $\mathbb{Z}_N$. A encriptação de $\Pi \in \{0, 1\}$ é dada pela expressão

$$\Psi = r^2 \times z^\pi \bmod N, \tag{4.12}$$

onde r é um número aleatório no intervalo $[0, N-1]$. Assim, caso $\pi = 0$, o texto cifrado é um resíduo quadrático aleatório em $\mathbb{Z}_N$. Por outro lado, se $\pi = 1$, o texto cifrado não é um resíduo quadrático. Assim, para decriptar os textos cifrados, basta descobrir se o valor é um resíduo quadrático ou não. Com relação ao homomorfismo,

esse esquema permite realizar adições de bits encriptados em módulo 2, ou seja, a função ou-exclusivo (XOR) pela multiplicação de textos cifrados,

$$\Psi_1 \times \Psi_2 = (r_1^2 z_1^{\pi}) \times (r_2^2 z_1^{\pi}) = z^{\pi_1+\pi_2}(r_1 r_2)^2 = ENC(\pi_1 \oplus \pi_2), \tag{4.13}$$

onde $\oplus$ é a operação de adição em módulo 2, ou seja, a função XOR. Portanto, o esquema de Goldwasser e Micali é parcialmente homomórfico para operações de adições e semanticamente seguro. No entanto, para prevenir ataques sobre a fatoração de N, o esquema requer N da ordem de algumas centenas de bits. Isso faz com que os textos cifrados sejam grandes em relação às suas mensagens originais. Portanto esse esquema é tido como uma prova de conceito e não foi utilizado na prática.

Seguiram ao criptossistema de Goldwasser e Micali, diversos outros sistemas de encriptação que são homomórficos para a adição ou para a multiplicação como, por exemplo, o esquema de encriptação de El Gamal [76], o esquema de Paillier [77] e sua generalização feita por Damgard e Jurik [78] e um conjunto de esquemas baseados em reticulados (*lattices*) inicialmente propostos por Ajtai e Dwork [79]. A principal restrição de todos esses sistemas de encriptação é a falta de suporte para a realização de operações homomórficas para adições e multiplicações ao mesmo tempo. A busca por esquemas homomórficos com essa propriedade é importante, pois, com adição e multiplicação, pode-se realizar um conjunto completo de operações, visto que a adição em módulo 2 é a operação de Ou-exclusivo (XOR) e a multiplicação módulo 2 é a operação de E (AND).

O esquema de Paillier [77] é parcialmente homomórfico para operações de adição. Sua segurança também é baseada no problema de fatoração de um número $N = p \times q$, com p e q primos grandes escolhidos aleatoriamente e com a mesma quantidade de bits, o que é garantido se $\gcd(pq, (p-1)(q-1)) = 1$. Nesse contexto, a chave privada é definida por $(\phi(N), \phi(N)^{-1})$, em que $\phi(N) = (p-1)(q-1)$, e a chave pública por $(N, N+1)$. A encriptação de uma mensagem $\pi \in \mathbb{Z}_N^*$ é dada pela expressão

$$\Psi = (N+1)^{\pi} \times r^N (mod\ N^2), \tag{4.14}$$

onde r é um número aleatório em que $r \in \mathbb{Z}_N^*$. A decriptação, por conseguinte, é realizada pela expressão

$$\pi = \frac{\left[\Psi^{\phi(N)}(mod\ N^2)\right] - 1}{N} \times \phi(N)^{-1} (mod\ N^2). \tag{4.15}$$

Dessa forma, com base na encriptação, esse esquema é homomórfico devido às pro-

priedades da exponenciação, como segue:

$$\begin{aligned}\Psi_1 \times \Psi_2 &= (N+1)^{\pi_1} r_1^N (N+1)^{\pi_2} r_2^N (mod\ N^2) \\ &= (N+1)^{\pi_1+\pi_2} (r_1 r_2)^N (mod\ N^2) \\ &= ENC(K_{pub}, \pi_1 + \pi_2).\end{aligned} \tag{4.16}$$

Portanto, o esquema é homomórfico para operações aditivas. Esse esquema permite ainda a multiplicação de um texto cifrado por uma mensagem em claro de forma homomórfica, o que é feito da seguinte forma:

$$\begin{aligned}\Psi_1^{\pi_2} &= ((N+1)^{\pi_1} r_1^N)^{\pi_2} (mod\ N^2) \\ &= (N+1)^{\pi_1 \pi_2} (r_1^{\pi_2})^N (mod\ N^2) \\ &= ENC(K_{pub}, \pi_1 \times \pi_2).\end{aligned} \tag{4.17}$$

As principais aplicações para o esquema de Paillier são o voto eletrônico e o dinheiro eletrônico.

Uma generalização do esquema de Paillier é proposta por Damgard e Jurik [78], a qual é realizada em aritmética modular de módulo N^{s+1}, em que N é o módulo do esquema RSA e s é um número natural positivo. Nesse contexto, o esquema de Paillier é um caso particular com $s = 1$. A principal vantagem desse esquema sobre o de Paillier é a habilidade de controlar os espaços de mensagens e de textos cifrados que podem ser encriptados com a chave pública. Especificamente, para uma dada chave pública, os espaços de mensagens e textos cifrados podem ser Z_{N^s}, $Z_{N^{s+1}}$, e assim sucessivamente para qualquer s. Isso permite a encriptação de mensagens arbitrariamente grandes.

Levieil e Naccache [80] propuseram uma técnica homomórfica com propriedades aditivas, com o objetivo de prevenir estudantes de trapacearem em suas provas. No entanto, esse esquema criptográfico não é seguro, independentemente da escolha dos parâmetros, pois os autores consideraram um modelo fraco de atacante[2] para demonstrar o esquema como é argumentado por Van Dijk *et al.* [1]. Isso é justificado porque, antes dos trabalhos de Craig Gentry [19] e van Dijk *et al.* [1], a comunidade científica considerava essa técnica insegura por natureza e, por isso, modelos fortes de atacante[3] não eram considerados. Em 2004, o primeiro esquema de encriptação de Regev [81] foi proposto baseado no problema do menor vetor único. A expressão de encriptação desse esquema é dada por

$$ENC(K_{priv}, m) = K_{priv} * q + 2r + m, \tag{4.18}$$

[2]Levieil e Naccache [80] assumem inclusive que os atacantes não conhecem a base matemática e o algoritmo de encriptação. Assim, a única alternativa é a força bruta sobre todas as possibilidades.

[3]Os atacantes fortes conhecem o algoritmo e toda a base matemática. Portanto, eles podem buscar métodos de reduzir o espaço de busca.

onde q e r são números aleatórios e m é a mensagem. No entanto, a chave privada é calculada pela razão entre o espaço de domínio N e um segredo h, isto é, $K_{priv} = N/h$. Ao contrário do esquema de Gentry e do esquema DGHV, as propostas de Levieil e Naccache e de Regev não oferecem o homomorfismo multiplicativo.

A construção de esquemas homomórficos que vão além de operações de adição ou de multiplicação isoladamente demorou certo tempo para acontecer. Boneh, Goh e Nissim [82] mostraram um esquema de encriptação baseado em emparelhamentos bi-lineares (*bilinear pairings*) sobre curvas elípticas [83] que é capaz de computar fórmulas quadráticas, ou seja, uma quantidade arbitrárias de adições assim como uma simples multiplicação sobre as mensagens originais. Após, Gentry, Halevi e Vaikuntanathan [84] apresentaram como alcançar essa mesma habilidade através de reticulados. O trabalho de Sander, Young e Yung [85] construiu um esquema capaz de avaliar circuitos polinomiais, pois o texto cifrado cresce exponencialmente com a profundidade do circuito. O esquema de Aguilar-Melchor, Gaborit e Herranz [86] avalia polinômios com múltiplas variáveis. Nesse esquema, o texto cifrado cresce exponencialmente com o grau do polinômio. A proposta de Fellows e Koblits [87] para um esquema totalmente homomórfico baseado na dificuldade do problema da associação ideal em anéis polinomiais $R = \mathcal{F}_q[x_1, x_2, \cdots, x_n]$. Essa proposta sofreu diversos ataques devido aos valores definidos para os parâmetros e a segurança do esquema ainda é um problema em aberto.

Esquemas de encriptação homomórficos para adição são úteis em diversas aplicações. Cohen e Fischer [88] propuseram um esquema de encriptação homomórfico para adição e mostraram como usá-lo para votações eletrônica seguras. Essa proposta e as demais originárias dessa proposta tiveram grande importância em sistemas de votação eletrônica baseados na Web, como o sistema Helios [89]. Outra aplicação é para protocolos de recuperação de informações privadas (PIR - *Private Information Retrieval*). A aplicação PIR foi definida por Chor, Kushilevitz, Goldreich e Sudan [90] e consiste no problema no qual o usuário tenta recuperar o i-ésimo item de uma base de dados de tamanho N sem revelar ao dono da base o índice i. Uma solução trivial é o dono da base enviar toda a base para o usuário, que escolhe o dado que deseja. Uma importante contribuição para a aplicação de PIR foi recebida do trabalho de Kushilevitz e Ostrovsky [91], que mostrou como construir um protocolo para recuperação de informações privadas de um servidor único a partir de qualquer esquema de encriptação homomórfico. Esse protocolo possui complexidade de comunicação sublinear, ou seja, o protocolo requer a comunicação de apenas N^{ϵ} bits, para uma base de dados de tamanho N e qualquer constante arbitrariamente pequena $\epsilon > 0$. No trabalho de Kushilevitz e Ostrovsky, os autores mostram que, para um parâmetro de segurança poli-logarítmico em N, a complexidade de comunicação é $\sqrt{log_N \; log \; log_N}$, ou seja, $0 < \epsilon < 1$, melhorando a solução trivial drasticamente. Um

grande desafio para a computação homomórfica ainda era a computação de desvios nos programas, como testes condicionais sobre os textos cifrados. Ishai e Paskin [92] mostraram como usar um esquema de encriptação homomórfica específico, mas bem eficiente, para construir um esquema capaz de avaliar homomorficamente decisões de fluxos em um programa.

Para resumir o estado da arte antes da proposta de Craig Gentry, em 2009, sabe-se que os esquemas de encriptação podiam realizar adições e uma simples multiplicação [82, 84] e outros esquemas podem computar programas de decisões [92], funções polinomiais [86] e circuitos NC^1 com expansão no tamanho do texto cifrado [85]. A grande contribuição na área surgiu com a proposta de Gentry [19], em 2009, a qual mostrou a primeira construção de um esquema totalmente homomórfico que possibilita a computação de funções arbitrárias sobre os dados encriptados, produzindo textos cifrados compactos.

4.4 A Proposta de Gentry

Craig Gentry [19] fez uma importante contribuição para o campo da criptografia homomórfica ao provar que era possível construir um esquema de encriptação totalmente homomórfica. A proposta de Gentry permite a execução de funções arbitrárias sobre os textos cifrados, obtendo-se a encriptação do resultado. Nesse método, a segurança é baseada no ruído adicionado ao texto cifrado. Quando se realiza uma operação, o ruído associado com o texto cifrado sofre a mesma operação. Então, existe um limite prático no número de operações que podem ser realizados porque o ruído vai se acumulando e aumentando a medida que as operações são realizadas. Eventualmente, após algumas operações, o ruído pode ficar tão grande que torna o texto cifrado impossível de ser decriptado. Então, Gentry propôs aplicar o algoritmo de decriptação sobre o texto cifrado de forma homomórfica sempre que o ruído pudesse tornar-se grande o suficiente para impossibilitar a decriptação. Aplicar a função de decriptação de forma homomórfica significa utilizar a encriptação da chave privada com outro par de chaves como chave de decriptação, como em

$$\Psi_2 = DEC(ENC(K^2_{pub}, K_{priv}), \Psi), \tag{4.19}$$

onde K^2_{pub} é a chave pública do novo par de chaves, K_{priv} é a chave privada do par de chaves que foi utilizado para se obter Ψ, que, por sua vez, é o texto cifrado atual. Assim, como a função de decriptação remove todo o ruído já presente no texto cifrado, a utilização homomórfica do algoritmo de decriptação pode provocar a redução do ruído. O ruído final consiste apenas no ruído introduzido pelas operações do algoritmo de decriptação, visto que essas operações são realizadas homomorficamente.

Gentry propôs, então, que esse procedimento fosse utilizado sempre que necessário para possibilitar a aplicação de infinitas operações homomórficas sobre os textos cifrados. Gentry chamou a operação de auto-inicialização (*bootstrapping*).

Mesmo com todos os seus benefícios e conceitos inovadores, a proposta de Gentry possui altos requisitos de processamento e armazenamento para ser usada em ambientes que requerem nível de segurança adequado. A fim de simplificar o esquema de encriptação, Van Dijk *et al.* [1] propuseram um esquema de encriptação homomórfica baseado em aritmética modular de inteiros. Van Dijk *et al.* desenvolveram um simples e seguro esquema de encriptação totalmente homomórfica, que também se utiliza do conceito de auto-inicialização proposto por Gentry.

4.5 A Proposta de Van Dijk *et al.*

Pela definição de um esquema de encriptação homomórfica definido na Seção 4.2.2, a construção desse esquema requer a definição dos quatro algoritmos (*Geração de Chaves*, *Encriptação*, *Decriptação* e *Avaliação*). Esta seção define cada um dos algoritmos para o esquema DGHV [1] bem como uma análise dos parâmetros considerando o que significam e os valores atribuídos para cada um deles. Sobre esse esquema, apresenta-se no Capítulo 5 a extensão proposta para cálculo de operações aritméticas de números inteiros grandes.

Definição dos Parâmetros

A construção do esquema especifica cinco parâmetros, os quais dependem polinomialmente do parâmetro de segurança λ. Quanto maior o parâmetro de segurança, maior a segurança do esquema. Cada um dos parâmetros possui uma função:

- η define o tamanho da chave privada em bits;
- γ representa, também em bits, o tamanho dos elementos da chave pública;
- τ determina a quantidade de elementos que compõem a chave pública;
- ρ define o tamanho em bits do ruído acrescido a cada um dos elementos da chave pública e
- ρ' especifica o tamanho do ruído adicionado ao texto cifrado no algoritmo de encriptação.

Van Dijk *et al.* [1] escolheram os valores dos parâmetros η, γ, ρ, ρ' e τ considerando uma série de ataques, como força bruta sobre o ruído adicionado e sobre o

problema do Máximo Divisor Comum Aproximado. Por isso, definem-se as relações abaixo para cada parâmetro em função do parâmetro de segurança λ:

$$\eta = \lambda^2; \tag{4.20a}$$

$$\gamma = \lambda^5; \tag{4.20b}$$

$$\tau = \gamma + \lambda + 1; \tag{4.20c}$$

$$\rho = \lambda; \tag{4.20d}$$

$$\rho' = 2\lambda. \tag{4.20e}$$

Ambos os parâmetros ρ e ρ' são escolhidos para dar robustez contra ataques de força bruta sobre os ruídos adicionados, respectivamente aos elementos da chave pública e aos textos cifrados. O parâmetro η é definido considerando a corretude do esquema, dado que ele determina o tamanho da chave privada, que será usada para o algoritmo de Decriptação. Além disso, o tamanho da chave privada também é importante para dar robustez contra ataques de força bruta para decriptar os valores. Os parâmetros γ e τ estão relacionados à redução para o problema do Máximo Divisor Comum Aproximado. Nesse contexto, enquanto o parâmetro γ é escolhido para prevenir ataques baseados em reticulados, o parâmetro τ é definido para possibilitar a aplicação do *Leftover Hash Lemma* [93] na redução para o problema do Máximo Divisor Comum Aproximado [1].

Geração de Chaves (λ)

O algoritmo de *Geração de Chaves* para o esquema de encriptação DGHV [1] pode ser utilizado para a criptografia simétrica ou para a criptografia assimétrica. Em ambos os casos, deve-se definir uma chave privada ou secreta. Para o caso assimétrico, a chave pública consiste em diversas encriptações do número zero.

A chave privada, K_{priv}, é um número inteiro ímpar, escolhido aleatoriamente no intervalo $[2^{\eta-1}, 2^{\eta})$. O parâmetro η determina o tamanho da chave privada em bits. Assim, como $\eta = \lambda^2$ [1], o tamanho da chave privada aumenta polinomialmente com o aumento do parâmetro de segurança λ.

No caso assimétrico, a chave pública, K_{pub}, consiste em $\tau + 1$ elementos $x_i = \mathcal{D}_{\gamma,\rho}(K_{priv})$, no qual a distribuição $\mathcal{D}_{\gamma,\rho}(K_{priv})$ é dada por

$$\mathcal{D}_{\gamma,\rho}(K_{priv}) = \{x_i = K_{priv} \times q_i + r_i | q_i \in [0, 2^{\gamma}/K_{priv}), r_i \in (-2^{\rho}, 2^{\rho})\}. \tag{4.21}$$

Após sortear todos os elementos da chave pública, reposicione os elementos x_i de forma que o elemento x_0 seja o maior dentre os elementos x_i. Além disso, o elemento x_0 deve ser ímpar e o resto da sua divisão pela chave privada (x_0/K_{priv}) deve ser

par. Nesse caso, o ruído do elemento x_0 é par, característica fundamental para a decriptação. Se o elemento x_0 não atender a essas restrições, deve-se reiniciar o procedimento até que se consiga um elemento x_0 com essas características. Ao fim do algoritmo, a chave pública é definida pela lista de inteiros $K_{pub} =< x_0, x_1, \cdots, x_\tau >$.

Encriptação (K_{pub}, π)

Para encriptar uma mensagem $\pi \in \{0, 1\}$, escolhe-se um subconjunto aleatório $S \subseteq \{1, 2, \cdots, \tau\}$, no qual $\tau + 1$ representa o número de elementos na chave pública. Sorteia-se ainda um número R inteiro no intervalo $(-2^{\rho'}, 2^{\rho'})$. Após a obtenção desses valores, o texto cifrado é computado através da expressão

$$\Psi = (\pi + 2R + 2\sum_{i \in S} x_i) \bmod x_0. \tag{4.22}$$

O texto cifrado Ψ pode ser usado pelo algoritmo de *Avaliação* para realizar as operações de adição e multiplicação homomorficamente.

Decriptação (K_{priv}, Ψ)

A decriptação do texto cifrado Ψ é obtida através da expressão

$$\pi' = (\Psi \bmod K_{priv}) \bmod 2. \tag{4.23}$$

Para a decriptação, o segredo está na paridade dos elementos. O resto da divisão do texto cifrado pela chave privada $(\Psi \bmod K_{priv})$ possui a mesma paridade do bit encriptado, pois essa operação elimina todos os termos da forma $K_{priv} \times q_i$, que possuem paridade desconhecida. Após a primeira divisão, restam apenas os elementos do ruído adicionado, os quais estão multiplicados por 2. Portanto, ao calcular o resto da divisão por 2, remove-se todo o ruído de encriptação, uma vez que os termos da soma $2\sum\ r_i$, o ruído de encriptação $2r$ são pares. Além disso, devido às suas restrições, o ruído do x_0 também é par. Esses fatores proporcionam a correta decriptação uma vez que a única forma de remover todo o ruído introduzido na encriptação é pelo uso da paridade.

Avaliação $(\Phi, < \Psi_1, \Psi_2, \cdots, \Psi_k >)$

Dado um circuito binário de operações Φ com k entradas e k textos cifrados Ψ_i, constrói-se um algoritmo que executa Φ sobre os textos cifrados. Por exemplo, para processar uma adição, simplesmente somam-se todos os textos cifrados

$< \Psi_1, \Psi_2, \cdots, \Psi_k >$ como no somatório

$$Resultado = \sum_{i}^{k} \Psi_i. \tag{4.24}$$

Desde que respeitado o tamanho máximo do circuito, qualquer circuito permitido pode ser utilizado com o algoritmo de *Avaliação*. Para o caso dos esquemas homomórficos apresentados nessa dissertação, são admitidos circuitos com as operações de adição e multiplicação. Vale ressaltar que, caso o circuito seja permitido para o esquema de encriptação $\mathfrak{E}$, a decriptação do resultado deve ser igual ao resultado do circuito ao aplicar as mensagens originais de cada um dos textos cifrados Ψ_i.

4.6 Estado da Arte

Devido ao tamanho da chave pública, armazenamento é um dos maiores desafios na encriptação homomórfica. No esquema DGHV, por exemplo, o tamanho da chave pública é da ordem de λ^{10} bits e da chave privada é λ^2 bits, no qual λ é o parâmetro de segurança. O tamanho da chave privada é necessário para garantir a segurança do esquema. Coron, Mandal e Naccache [94] propuseram uma modificação na geração das chaves com o objetivo de reduzir o tamanho da chave pública para λ^7 bits. A modificação consiste em usar formas quadráticas dos elementos da chave pública ao invés de formas lineares, como é feito no esquema DGHV. A ideia de Coron *et al.* consiste em armazenar apenas um pequeno subconjunto da chave pública e, quando necessário, gerar a chave pública completa combinando os elementos do subconjunto multiplicativamente. Esta proposta mantém a segurança semântica, pois é baseada no problema do máximo divisor comum parcial aproximado (*partial approximate greatest common divisor*) [95], que consiste em retirar o erro do primeiro termo da chave pública ($x_0 = q_0 \times K_{priv}$). Esse problema possui a mesma base de segurança do problema do máximo divisor comum aproximado (*approximate greatest common divisor*), que é a base da segurança do esquema DGHV. Outra proposta com o mesmo objetivo de reduzir o tamanho da chave pública é a de Coron, Naccache e Tibouchi [96], que apresenta algumas otimizações adicionais para o tamanho da chave pública, reduzindo-o para λ^5 bits. Ambas as propostas são baseadas no esquema DGHV e focam na encriptação de bits. Com esse mesmo objetivo, porém aplicado ao método original de Gentry, Smart e Vercauteren [20] propuseram um novo método para geração do par de chaves e para encriptação, o qual reduz tanto o tamanho do texto cifrado quanto o tamanho da chave pública. Além disso, na proposta de Smart e Vercauteren, os autores defendem que o esquema proposto é capaz de encriptar K bits de uma só vez ao contrário da proposta original de Gentry, que requer a decomposição da mensagem em bits para o posterior processamento

bit a bit. Essa capacidade de processamento de múltiplos bits por operação está na mesma direção da proposta nessa dissertação. Contudo, a proposta dessa dissertação foca em calcular sobre os números inteiros, diretamente sobre uma extensão do esquema DGHV, que é mais simples que o esquema de Gentry.

Gentry e Halevi [97] apresentaram uma implementação do esquema de encriptação original de Gentry [19] incluindo a operação de auto-inicialização (*bootstrapping*). Os autores analisaram o desempenho da implementação considerando os algoritmos básicos (geração de chaves, encriptação e decriptação), a operação de auto-inicialização e o tamanho da chave pública. Com o objetivo de melhorar o desempenho do esquema original do Gentry, Brakerski e Vaikuntanathan [98] propuseram um esquema de encriptação baseado na premissa do problema de aprendizado de máquina conhecido como *Learning With Errors* (LWE - Aprendendo com Erros). A segurança de esquemas criptográficos fundamentados em problemas LWE é baseada na dificuldade do pior caso do problema do vetor mínimo em reticulados arbitrários [99]. Essa mudança de paradigma e a introdução de uma operação para redução da dimensão dos vetores produz textos cifrados menores, o que aumenta o desempenho da decriptação. Além disso, a aplicação dessas técnicas baseadas em LWE levaram ao trabalho de Brakerski *et al.* [100], que construiu um esquema de encriptação totalmente homomórfico para operação sobre circuitos polinomiais sem requerer a operação de auto-inicialização. No trabalho de Brakerski *et al.*, a segurança é baseada no problema LWE em anéis (RLWE), que foi recentemente apresentado por Lyubashevsky *et al.* [101].

4.7 Desafios da Criptografia Homomórfica

O primeiro desafio para a criptografia homomórfica é sua viabilidade prática. Enquanto a construção original de Gentry [19] é vista como não prática, construções recentes e esforços de implementacão têm melhorado drasticamente a eficiência da encriptação totalmente homomórfica. Esforços iniciais de implementação, focados na proposta original de Gentry e suas variantes [20, 94, 97, 102] parecem melhorar gargalos de eficiência da proposta original. Implementações posteriores fazendo uso de avanços recentes nos algoritmos [98, 100, 103] e técnicas algébricas [102, 104] para melhorar a eficiência concreta dos esquemas resultam em esquemas de encriptação totalmente homomórfica com menos restrições para uso.

Outra questão é a contraposição entre as propriedades de homomorfismo e não maleabilidade de um esquema criptográfico. A encriptação homomórfica permite que qualquer um possa transformar a encriptação de uma mensagem π na encriptação de $\Phi(\pi)$ para funções Φ não triviais. Encriptação não maleável, por outro lado, consiste na prevenção desse tipo de operação. Essa propriedade requer que nenhum

adversário seja capaz de transformar a encriptação de π na encriptação de qualquer outra mensagem relacionada. Na prática, o algoritmo de *Avaliação* deve ser capaz de computar homomorficamente qualquer função de uma classe pré-especificada $\mathcal{F}_{hom}$ e não conseguir transformar a encriptação de π na encriptação de $\Phi(\pi)$ para qualquer $\Phi \notin \mathcal{F}_{hom}$. Então, a questão que se faz é se é possível controlar o que está sendo computado homomorficamente. Boneh, Segev e Waters [105] propuseram a noção de maleabilidade alvo (*targeted malleability*), que é uma formalização candidata para esse requisito. O esquema de encriptação proposto por eles permite uma avaliação iterativa de até t funções, onde t é uma constante pré-especificada. Melhorar a construção proposta por Boneh *et al.* assim como as premissas de complexidade usadas ainda é um importante problema em aberto.

Além disso, é interessante estender a definição de não maleabilidade para considerar os ataques de texto cifrado escolhidos (CCA). Considerando, por exemplo, a implementação de um sistema de recomendação de filmes que gera as suas recomendações com base no conteúdo do e-mail dos usuários, pode-se construir um esquema baseado na encriptação homomórfica. Uma vez que o e-mail é armazenado encriptado com a chave pública do usuário, o servidor realiza uma avaliação homomórfica e computa uma recomendação enviada para o usuário. O usuário descriptografa a recomendação recebida e realiza uma ação de acordo com os seus interesses em filmes. De fato, se a recomendação for relevante para o usuário, ele deve escolher clicar para obter mais informações e, em caso contrário, o usuário ignora a recomendação. Assim, considere que se o servidor de e-mail for capaz de saber a ação tomada pelo usuário, ele pode utilizar esse sistema de recomendação como um oráculo de decriptação para quebrar a segurança do esquema usado pelo usuário ou, talvez, descobrir a chave privada do usuário. Esse tipo de ataque pode acontecer em qualquer aplicação em que se computa sobre dados encriptados, o que indica que a segurança contra ataques de texto cifrado escolhido (*CCA-security*) parece inviável. Com esse exemplo, é possível concluir que esquemas de encriptação homomórfica seguros contra ataques CCA2 não podem existir.

Portanto, conclui-se que muitos desafios ainda estão em aberto considerando a encriptação homomórfica e suas implicações para a segurança dos esquemas, mas os avanços apresentados em trabalhos recentes faz com que a técnica esteja cada vez mais próxima de ambientes práticos reais.

Capítulo 5

Proposta de Esquema de Encriptação Homomórfica com Operações sobre Números Inteiros Grandes

Essa dissertação propõe um novo esquema de encriptação orientado ao cálculo de funções aritméticas de números inteiros grandes [106]. A proposta é uma extensão do esquema de encriptação homomórfica limitado (DGHV), de Van Dijk *et al.* [1]. A limitação do esquema consiste no número de operações que podem ser computadas com o texto cifrado antes de ocorrer erro na decriptação do resultado. O esquema DGHV foca na encriptação e no processamento de bits. Embora seja possível processar pequenos números inteiros, dependendo do tamanho da chave pública escolhida, o esquema DGHV não é indicado para processamento de números de tamanho arbitrário. Para números grandes, o esquema DGHV requer o mapeamento dos números e das operações para adições e multiplicações binárias. O objetivo desta dissertação é melhorar a eficiência do esquema DGHV para operações aritméticas sobre números inteiros. Assim, com o objetivo de permitir operações sobre números inteiros grandes sem processá-los bit a bit, introduz-se um novo parâmetro chamado Base, ou simplesmente, B. Esse novo parâmetro representa a quantidade máxima de valores diferentes que o par de chaves escolhido é capaz de encriptar, processar e decriptar. Portanto, o esquema DGHV pode ser visto como um caso particular do esquema proposto, em que $B = 2$.

Neste Capítulo, a Seção 5.1 apresenta os algoritmos de *Geração de Chaves*, *Encriptação* e *Decriptação* da extensão proposta. A Seção 5.2 apresenta a análise de corretude do esquema, incluindo análises para as operações de adição e de multiplicação, enquanto a Seção 5.3 apresenta uma análise das limitações na profundidade

do circuito que pode ser computado com o esquema e uma análise dos tamanhos das chaves pública e privada do esquema, considerando diversos valores para o parâmetro de segurança (λ) e para a base do esquema (B). Com base nas limitações encontradas, a Seção 5.4 apresenta uma discussão informal sobre a expansão do esquema proposto para que seja um esquema completamente homomórfico. A Seção 5.5 discute a segurança semântica do esquema proposto com base nas premissas adotadas pelo esquema DGHV. E, por fim, a Seção 5.6 define as métricas de avaliação e os resultados de uma análise experimental sobre a extensão proposta além de apresentar alguns detalhes de implementação da extensão.

5.1 Definição da Proposta

O esquema DGHV, descrito na Seção 4.5, é restrito para operações binárias. Portanto, o esquema DGHV requer um alto número de operações para calcular funções aritméticas simples sobre números inteiros grandes. Logo, com o objetivo de permitir a encriptação e o processamento de números inteiros de tamanho arbitrário em um número reduzido de simples operações, essa dissertação propõe a introdução do parâmetro B (BASE), o qual representa a quantidade de números diferentes que um par de chaves é capaz de encriptar e decriptar. Por conseguinte, deve-se generalizar o conceito de paridade usado no esquema DGHV para o conceito de divisibilidade. Por essa razão, requer-se que o ruído aplicado utilizado seja múltiplo de B ao invés de ser par. De todo modo, o esquema DGHV é um caso particular da extensão proposta, no qual $B = 2$. Analogamente, quando o esquema original requer números ímpares, a extensão proposta requer números não múltiplos de B.

A proposta também aumenta a segurança do sistema contra ataques de força bruta, uma vez que com o aumento da base B, a proporção de números não múltiplos em relação aos números múltiplos aumenta e, portanto, há mais possibilidades para a escolha da chave privada K_{priv}. No restante dessa seção, apresentam-se as modificações em relação ao esquema DGHV no que concerne a definição dos parâmetros e a construção dos três algoritmos básicos. O algoritmo de *Avaliação*, no entanto, é mantido inalterado, visto que consiste apenas na execução do circuito de operações Φ sobre os textos cifrados Ψ_i usados como entrada para o circuito.

Definição de Parâmetros da Proposta

Com o objetivo de preservar a segurança semântica do esquema e da sua redução para o problema do Máximo Divisor Comum Aproximado, a maioria dos parâmetros são definidos da mesma forma que foram definidos no esquema DGHV e apresentados na Equação 4.20. A única exceção é o parâmetro τ, o qual deve ser redefinido como

$\tau = \lambda^5 + \lambda + log_2\ B$. Observe que essa definição de τ é uma extensão da definição do parâmetro expressa na Equação 4.20c, pois com $B = 2$, $log_2\ 2 = 1$. Essa modificação é necessária para garantir a indistinguibilidade estatística dos textos cifrados considerando um ataque que tenta prever o bit menos significativo da chave privada, mantendo, ao menos, o mesmo nível de segurança do esquema DGHV [1]. Portanto, em resumo, os parâmetros usados na extensão proposta são definidos como:

$$\eta = \lambda^2; \tag{5.1a}$$

$$\gamma = \lambda^5; \tag{5.1b}$$

$$\tau = \gamma + \lambda + log_2\ B; \tag{5.1c}$$

$$\rho = \lambda; \tag{5.1d}$$

$$\rho' = 2\lambda. \tag{5.1e}$$

Geração de Chaves (λ)

A chave privada K_{priv} é um número indivisível por B escolhido aleatoriamente no intervalo $[B^{\eta-1}, B^{\eta})$. Dessa forma, observa-se que o tamanho da chave privada cresce polinomialmente com λ vezes o logaritmo em base 2 de B, ou seja, $Tamanho(K_{priv}) \approx \lambda^2 \times log_2\ B$.

Similarmente ao esquema DGHV, a chave pública K_{pub} é composta de τ elementos. Cada elemento da chave pública é sorteado obedecendo a distribuição $\mathcal{D}_{\gamma,\rho}(K_{priv})$, que na extensão proposta é dada por

$$\mathcal{D}_{\gamma,\rho}(K_{priv}) = \{x_i = K_{priv} \times q_i + r_i | q_i \in [0, B^{\gamma}/K_{priv}), r_i \in (-B^{\rho}, B^{\rho})\}. \tag{5.2}$$

Essas modificações nos intervalos de busca dos valores de q_i e de $r_i q$ são fundamentais para garantir as mesmas premissas assumidas no esquema DGHV.

Após o sorteio dos elementos da chave pública, devem-se reposicionar os elementos x_i de forma que o elemento x_0 seja o maior dentre os elementos x_i. Além disso, o elemento x_0 não deve ser divisível pela base B e o resto da sua divisão pela chave privada, que é igual a x_0/K_{priv}, deve ser divisível por B. Essas restrições obrigam que o ruído do elemento x_0 seja divisível pela base B, ou seja, $r_0 \bmod B = 0$. Caso x_0 não atenda a essas restrições, reinicia-se o procedimento de geração da chave pública até que se consiga um elemento x_0 com essas características. Ao fim do algoritmo a chave pública é definida pela lista de inteiros $K_{pub} =< x_0, x_1, \cdots, x_{\tau} >$.

Encriptação (K_{pub}, π)

A encriptação de um inteiro $\pi \in [0, B)$, no qual B é a base utilizada no esquema $\mathfrak{E}$, é expressa por

$$\Psi = (\pi + B \times R + B \sum_{i \in S} x_i) \bmod x_0, \tag{5.3}$$

no qual o conjunto S é gerado aleatoriamente a partir dos elementos da chave pública e o ruído R é um número aleatório no intervalo $(-B^{\rho'}, B^{\rho'})$.

Decriptação (K_{priv}, Ψ)

Na extensão proposta, a expressão de decriptação é alterada para

$$\pi' = (\Psi \bmod K_{priv}) \bmod B. \tag{5.4}$$

As premissas de paridade que fundamentam a decriptação no esquema de DGHV são mantidas, porém sob a ótica do conceito de divisibilidade. Quando um número é divisível pela base definida no esquema, significa que ele seria par em uma base 2. Dessa forma, o cálculo do resto pela chave privada na primeira divisão $(\Psi \bmod K_{priv})$ remove toda a componente múltipla da chave privada do texto cifrado, restando apenas o ruído. Como o ruído é múltiplo da base B do esquema, o cálculo do resto na segunda divisão remove todo o ruído do texto cifrado, deixando apenas o valor da mensagem original π.

5.2 Análise de Corretude da Proposta

A corretude da decriptação é baseada no conceito da divisibilidade. A função de decriptação calcula o resto da divisão do texto cifrado pela chave privada e, depois, o resto da divisão por B. Dessa forma, se o objetivo é decriptar corretamente o resultado, deve-se garantir que o ruído seja divisível por B. Uma vez que o esquema é baseado na aritmética modular, se a soma dos ruídos for maior do que a metade da chave privada, então existe erro de decriptação, pois a mensagem original será adicionada pelo quociente da divisão do ruído por B.

Conforme exposto na Seção 5.1, dada uma mensagem $\pi \in \{\mathbb{Z} \mid 0 \leq \pi < B\}$, e a expressão de encriptação $\Psi = (\pi + B \times R + B \sum_{i \in S} x_i) \bmod x_0$, obtém-se o texto cifrado através da expressão

$$\Psi = \pi + B \times R + B \sum_{i \in S} x_i - k \times x_0, \tag{5.5}$$

na qual k é o quociente inteiro da divisão de $\pi + B \times R + B \sum_{i \in S} x_i$ por x_0. Como

é sabido que $x_i = K_{priv} \times q_i + r_i$, é possível reescrever o texto cifrado como

$$\Psi = \pi + B \times R + B \sum_{i \in S} r_i - k \times r_0 + K_{priv}(\sum_{i \in S} q_i - k \times q_0). \tag{5.6}$$

Para o algoritmo de decriptação, utiliza-se a expressão $\pi' = (\Psi \bmod K_{priv}) \bmod B$. Dessa forma, o texto cifrado, depois da primeira divisão da expressão de decriptação é dado por

$$\pi' = \pi + B \times R + B \sum_{i \in S} r_i - k \times r_0 - k' \times K_{priv}, \tag{5.7}$$

na qual k' é o quociente inteiro da divisão de Ψ na Equação 5.6 por K_{priv}. Assim, com objetivo de garantir a corretude da decriptação, todos os elementos da soma, exceto π, devem ser divisíveis por B. Caso isso seja verdade, com o cálculo do resto da divisão por B, todo o ruído é removido e recupera-se a mensagem original π.

Analisando-se cada um dos termos da expressão acima, têm-se:

- as parcelas $B \times R$ e $B \sum_{i \in S} r_i$ são claramente divisíveis por B;
- a parcela $k \times r_0$ é divisível por B devido às restrições na definição do elemento x_0. Como $x_0 = K_{priv} \times q_0 + r_0$ e as restrições determinam que o elemento x_0 não é divisível por B e que $x_0 \bmod K_{priv}$ é divisível por B, então significa dizer que r_0 é um número inteiro diferente de zero e divisível por B;
- quanto à parcela $-k' \times K_{priv}$, não há restrições que garantam a sua divisibilidade por B, pois garantidamente a chave privada não é divisível por B e o valor k' pode assumir qualquer valor. Nesse caso, deve-se garantir que esse termo é zero.

Portanto, a prova de corretude do esquema de encriptação requer que

$$-k' \times K_{priv} = 0, \tag{5.8}$$

o que acontece se, e somente se,

$$\pi + B \times R + B \sum_{i \in S} r_i - k \times r_0 < \frac{K_{priv}}{2}. \tag{5.9}$$

Essa condição é fundamental para que o quociente da divisão pela chave privada, realizada na decriptação, seja zero e o resto da divisão por B seja exatamente a mensagem encriptada. Devido a forma como a operação de resto é realizada no esquema DGHV [1], o limite superior é definido como $\frac{K_{priv}}{2}$. Tradicionalmente, a operação de divisão de números inteiros é feita com o quociente aproximado pelo inteiro menor,

gerando um resto positivo. No esquema DGHV, o quociente é aproximado para o inteiro mais próximo, podendo gerar um resto positivo ou negativo. Em ambas as abordagens, a prova real da divisão é satisfeita, ou seja,

$$dividendo = divisor \times quociente + resto. \tag{5.10}$$

No entanto, com a abordagem utilizada, o ruído no texto cifrado é menor. Dessa forma, para que $-k'$, na Equação 5.8, seja 0, o ruído deve ser menor do que a metade do divisor, que, no caso da primeria operação da Equação 5.4, é a chave privada K_{priv}. Assim, o resultado da decriptação é correto após o cálculo do resto por B, na operação seguinte.

Com isso, podem-se definir as condições de contorno para a corretude do esquema proposto em função do parâmetro de segurança λ e da base do esquema B. Como o esquema é baseado em sorteios aleatórios, a definição em função de λ e B é feita considerando-se o pior caso, o qual é provocaria maior soma total e menor chave privada possível. Analisando o pior caso, é possível determinar o ruído máximo suportado pelo esquema para que a decriptação seja feita corretamente. Para a análise do pior caso, utilizam-se as seguintes ordens de grandeza:

- $\pi \approx B$, pois $B - 1$ é o maior valor que π pode assumir;
- $R \approx B^{2\lambda}$, pois é o valor máximo de ρ';
- $r_i \approx B^{\lambda}$, pois é o valor máximo de ρ;
- $r_0 \approx -B^{\lambda}$, pois é o valor mínimo de ρ;
- $|S| \approx \lambda^5$, pois é o tamanho máximo do conjunto S, dado por τ;
- $k \approx |S|$, pois k será máximo quando todos os valores de x_i forem ligeiramente menores do que x_0.

Assim, a ordem de magnitude do ruído de encriptação é dada por

$$B \times B^{2\lambda} + B \times \lambda^5 \times B^{\lambda} - \lambda^5 \times (-B^{\lambda}). \tag{5.11}$$

Consequentemente, considerando apenas λ e B, no pior caso, a extensão proposta decripta o mensagem original corretamente se

$$B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right) < \frac{K_{priv}}{2}. \tag{5.12}$$

Portanto, para definir o limite de operações suportado pelo esquema proposto sem precisar da operação de auto-inicialização, é preciso analisar essa inequação para

determinar os limites mínimos no número de operações. Porém, antes da análise do limite de operações, apresenta-se a corretude da decriptação para textos cifrados resultados do algoritmo de *Avaliação* com circuitos aditivos e multiplicativos.

5.2.1 Corretude para Operações de Adição

O texto cifrado gerado pelo algoritmo de *Avaliação* para circuitos aditivos de duas parcelas é dado por

$$\Psi^{+} = \Psi^{I} + \Psi^{II}, \tag{5.13}$$

que, considerando a expressão de encriptação, é dado por

$$\Psi^{+} = (\pi^{I} + B \times R^{I} + B \sum_{i \in S^{I}} x_i) \bmod x_0^{I} + (\pi^{II} + B \times R^{II} + B \sum_{i \in S^{II}} x_i) \bmod x_0^{II}. \tag{5.14}$$

Como $x_0^{I} = x_0^{II} = x_0$, pode-se reescrever Ψ^{+} para

$$\Psi^{+} = (\pi^{I} + \pi^{II} + B \times R^{I} + B \times R^{II} + B \sum_{i \in S^{I}} x_i + B \sum_{i \in S^{II}} x_i) \bmod x_0. \tag{5.15}$$

Dessa forma, na função de decriptação, após a aplicação da primeira divisão, tem-se que π'^{+} é dado por

$$\pi'^{+} = \pi^{I} + \pi^{II} + B \times R^{I} + B \times R^{II} + B \sum_{i \in S^{I}} r_i + B \sum_{i \in S^{II}} r_i - k \times r_0 - k' \times K_{priv}. \tag{5.16}$$

Assim, pode-se observar que o ruído total é a soma dos ruídos de cada um dos textos cifrados que compuseram a soma. Portanto, de forma semelhante à análise apresentada para a decriptação da mensagem encriptada, para obter-se a correta decriptação do texto cifrado resultado da adição é preciso que, no pior caso, a inequação

$$\left[B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right)\right]^{I} + \left[B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right)\right]^{II} < \frac{K_{priv}}{2} \tag{5.17}$$

precisa ser válida.

Logo, para um circuito aditivo de profundidade d qualquer, a corretude do circuito é definida pela inequação

$$(d+1) \times \left[B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right)\right] < \frac{K_{priv}}{2}. \tag{5.18}$$

5.2.2 Corretude para Operações de Multiplicação

De forma análoga à análise desenvolvida para as operações de multiplicação, esta Seção apresenta a análise de corretude para circuitos multiplicativos. Nesse contexto, o texto cifrado gerado pelo algoritmo de *Avaliação* para circuitos multiplicativos de dois fatores é dado por

$$\Psi^* = \Psi^I \times \Psi^{II}, \tag{5.19}$$

que, considerando a expressão de encriptação, é dado por

$$\Psi^* = (\pi^I + B \times R^I + B \sum_{i \in S^I} x_i) \bmod x_0^I \times (\pi^{II} + B \times R^{II} + B \sum_{i \in S^{II}} x_i) \bmod x_0^{II}. \tag{5.20}$$

Procedendo, portanto, uma análise semelhante à aplicada aos circuitos aditivos, o ruído de encriptação após a operação de multiplicação, no pior caso, é dado pela inequação

$$\left[B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right)\right]^I \times \left[B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right)\right]^{II} < \frac{K_{priv}}{2}, \tag{5.21}$$

que precisa ser válida, para a correta decriptação do resultado. Por conseguinte, para um circuito multiplicativo de profundidade d qualquer, a corretude do circuito é definida pela inequação

$$\left[B + B^{\lambda+1}\left(B^2 + 2\lambda^5\right)\right]^{d+1} < \frac{K_{priv}}{2}. \tag{5.22}$$

5.3 Análise dos Limites dos Circuitos e dos Tamanho das Chaves

O esquema proposto tem um limite no número de operações que podem ser realizadas porque o ruído do texto cifrado aumenta com a execução de cada operação. Na Seção 5.2, com o objetivo de decriptar corretamente o texto cifrado, conclui-se que o ruído total deve ser menor do que a metade do valor da chave privada, como definido na Equação 5.9. Com base nisso, quando se efetua uma soma $(\Psi_1 + \Psi_2)$ ou uma multiplicação $(\Psi_1 \times \Psi_2)$, o ruído agregado sofre a mesma operação, conforme demonstrado nas Seções 5.2.1 e 5.2.2, respectivamente. Por essa razão é que existe o limite na profundidade do circuito a ser computado.

O limite específico de operações depende do ruído de cada termo e da operação a ser realizada. No entanto, para analisar o pior caso, podem-se considerar os ruí-

Tabela 5.1: Profundidade máxima dos circuitos e tamanho máximo das chaves públicas e privadas.

Base (B)	λ	K_{priv} (bytes)	K_{pub} (bytes)	Adições	Multiplicações
2	5	4	$1,22 \times 10^{6}$	19	1
	10	13	$1,19 \times 10^{9}$	$7,73 \times 10^{20}$	3
	20	50	$1,22 \times 10^{12}$	$4,80 \times 10^{106}$	9
	40	200	$1,25 \times 10^{15}$	$2,46 \times 10^{460}$	23
	80	800	$1,18 \times 10^{18}$	$6,16 \times 10^{1.891}$	56
	160	3.200	$1,31 \times 10^{21}$	$9,51 \times 10^{7.645}$	128
2^8	5	25	$9,8 \times 10^{6}$	1.55×10^{38}	2
	10	100	$1,0 \times 10^{10}$	$1,58 \times 10^{206}$	7
	20	400	$1,02 \times 10^{13}$	$1,59 \times 10^{903}$	16
	40	1.600	$1,04 \times 10^{16}$	$2,66 \times 10^{3.743}$	35
	80	6.400	$1,07 \times 10^{19}$	$1,38 \times 10^{15.205}$	75
	160	25.600	$1,09 \times 10^{22}$	$1,53 \times 10^{61.249}$	154
2^{16}	5	50	$1,96 \times 10^{7}$	$5,78 \times 10^{76}$	2
	10	200	$2,0 \times 10^{10}$	$8,24 \times 10^{413}$	7
	20	800	$2,04 \times 10^{13}$	$4,95 \times 10^{1.810}$	17
	40	3.200	$2,09 \times 10^{16}$	$1,32 \times 10^{7.494}$	37
	80	12.800	$2,14 \times 10^{19}$	$1,52 \times 10^{30.420}$	77
	160	51.200	$2,19 \times 10^{22}$	$9,66 \times 10^{122.509}$	156
2^{32}	5	100	$3,95 \times 10^{7}$	$6,7 \times 10^{153}$	2
	10	400	$4,0 \times 10^{10}$	$1,35 \times 10^{828}$	7
	20	1.600	$4,09 \times 10^{13}$	$4,91 \times 10^{3.621}$	17
	40	6.400	$4,19 \times 10^{16}$	$3,84 \times 10^{14.988}$	37
	80	25.600	$4,29 \times 10^{19}$	$2,97 \times 10^{60.841}$	77
	160	102.400	$4,39 \times 10^{22}$	$4,63 \times 10^{245.023}$	157
2^{64}	5	200	$7,98 \times 10^{7}$	$8,98 \times 10^{307}$	2
	10	800	$8,0 \times 10^{10}$	$3,69 \times 10^{1.656}$	7
	20	3.200	$8,19 \times 10^{13}$	$4.83 \times 10^{7.243}$	17
	40	12.800	$8,38 \times 10^{16}$	$2,95 \times 10^{29.977}$	37
	80	51.200	$8,58 \times 10^{19}$	$1,76 \times 10^{121.683}$	77
	160	204.800	$8,79 \times 10^{22}$	$4,29 \times 10^{490.047}$	157
2^{128}	5	400	$1,62 \times 10^{8}$	$1,61 \times 10^{616}$	2
	10	1.600	$1,6 \times 10^{11}$	$2,73 \times 10^{3.313}$	7
	20	6.400	$1,63 \times 10^{14}$	$4.68 \times 10^{14.487}$	17
	40	25.600	$1,67 \times 10^{17}$	$1,74 \times 10^{59.955}$	37
	80	102.400	$1,71 \times 10^{20}$	$6,26 \times 10^{243.366}$	77
	160	409.600	$1,75 \times 10^{23}$	$3,69 \times 10^{980.095}$	157

dos máximos para cada operação, conforme apresentado nas Equações 5.18 e 5.22. Na Tabela 5.1, apresenta-se os números máximos de operações suportadas por cada cenário para o pior caso. Dessa forma, garante-se que esse número de operações sempre é possível para todos os casos. Observa-se, assim, que o número de adições permitidas é bem maior que o número de multiplicações. Além disso, conforme a base, B, e o nível de segurança, λ, aumentam, maior é a profundidade do circuito que pode ser computado. A proposta desta dissertação de introdução do parâmetro B permite um aumento no número de adições com um pequeno aumento no tamanho das chaves públicas e privadas. No esquema DGHV original, para aumentar o número de operações é necessário aumentar o parâmetro de segurança λ, o que implica num aumento bem superior da chave pública. Por outro lado, o número de multiplicações se mantém inalterado conforme B aumenta, pois, segundo a Equação 5.22, o ruído aumenta exponencialmente com o número de multiplicações enquanto o ruído permitido cresce por um fator de $log_2\ B$. Portanto, o parâmetro B adicionado na proposta, além de possibilitar operações sobre números maiores, permite mais adições consecutivas enquanto não altera a quantidade de multiplicações possíveis.

A Tabela 5.1 também apresenta o tamanho máximo, em bytes, estimado para das chaves pública e privada para cada configuração de base (B) e parâmetro de segurança (λ) analisados. Para o cálculo do tamanho máximo, busca-se o valor máximo da chave e, posteriormente, calcula-se o tamanho em bits requerido para a representação do valor da chave. A estimativa de valor máximo da chave privada considera o valor máximo que a chave privada pode assumir como sendo o valor máximo do intervalo de sorteio da chave privada, ou seja, o valor máximo estimado é dado por

$$Max(K_{priv}) = B^{\lambda^2}. \tag{5.23}$$

Por outro lado, para a chave pública, considera-se que todos os elementos x_i tivessem valor máximo, ou seja, q_i e r_i sejam sorteados pelos valores máximos. Considera-se também que se utiliza a chave privada máxima. Assim, o tamanho máximo da chave pública é dado por

$$Max(K_{pub}) = (\tau + 1) \times \left[Max(K_{priv}) \times \frac{B^{\lambda^5}}{Max(K_{priv})} + B^{\lambda} \right]. \tag{5.24}$$

Dessa forma, pode-se avaliar o crescimento da chave pública e da chave privada com o aumento de B e λ. Conclui-se, portanto, que o aumento da chave pública é impeditivo para a aplicação prática, com níveis de segurança ideais, tanto do esquema DGHV original quanto da extensão proposta.

5.4 Discussão sobre Criptografia Totalmente Homomórfica

Embora o esquema DGHV mostrado na Seção 4.5 e a extensão proposta nessa dissertação sejam esquemas de encriptação homomórfica para adição e para multiplicação, a Seção 5.3 mostra que ambos são limitados no número de operações que podem ser realizadas. Uma solução para esse problema é adotar a operação de auto-inicialização (*bootstrapping*), proposta por Craig Gentry [19]. Essa operação almeja reduzir o ruído presente no texto cifrado quando for necessário e seu conceito fundamental é realizar a operação de decriptação homomorficamente para, depois, re-encriptar o texto cifrado.

No caso geral, a re-encriptação deve ser realizada com novo par de chaves. Caso seja possível e seguro encriptar a chave privada com a sua própria chave pública, pode-se então utilizar o mesmo par de chaves para a etapa de re-encriptação. Nesse cenário, considera-se que o esquema $\mathfrak{E}$ possui segurança circular [75]. No entanto, tanto o esquema DGHV apresentado quanto a extensão proposta não permitem a encriptação da chave privada, visto que a chave privada é sempre maior do que o valor da base B, que representa os números que podem ser encriptados com o esquema, pois a chave privada é definida no intervalo $[B^{\eta-1}, B^{\eta})$.

A operação de autoinicialização é poderosa porque permite o processamento de circuitos de qualquer profundidade, ou, para o cenário dos esquemas apresentados nessa dissertação, funções polinomiais de qualquer ordem. Por outro lado, essa operação requer que o algoritmo de decriptação seja processado homomorficamente, o que, na prática, significa que a profundidade do circuito de decriptação deve ser menor do que a profundidade máxima aceitável pelo esquema. Assim, pode-se considerar que a operação de autoinicialização é um caso particular da utilização do algoritmo de *Avaliação*, no qual a função avaliada é o próprio algoritmo de *Decriptação* e os parâmetros são o texto cifrado (Ψ) e a encriptação da chave privada ($ENC(K_{priv})$). Portanto, o novo texto cifrado (Ψ') perde todo o ruído original que estava presente em Ψ, mas um novo ruído é introduzido pela execução homomórfica do algoritmo de *Decriptação*.

No entanto, no esquema DGHV apresentado e na extensão proposta, o circuito de decriptação requer mais operações do que o esquema aceita. Por essa razão, Van Dijk *et al.* [1] propuseram mudanças no esquema com o objetivo de reduzir a profundidade do circuito de decriptação. De fato, a extensão proposta aumenta a complexidade do circuito de decriptação, pois opera com números maiores. Então, as vantagens de aplicar as modificações propostas nessa dissertação para o esquema totalmente homomórfico proposto por Van Djik *et al.* não estão claras. Nessa dissertação, analisa-se o limite máximo de operações que o esquema proposto aceita,

mas continua em aberto uma análise formal sobre a operação de auto-inicialização sobre a extensão proposta. Essa análise deve focar na verificação da viabilidade de tornar a extensão totalmente homomórfica. O objetivo principal da proposta da dissertação é apresentar um esquema que permita operações simples sobre números inteiros grandes. Mesmo com um limite no número de operações, diversas aplicações práticas são atendidas pela proposta, como transações bancárias, computações multiparte e agentes móveis.

5.5 Discussão sobre a Segurança do Esquema Proposto

Considerando ataques contra um esquema de encriptação assimétrico, ataques que buscam descobrir a chave privada a partir da chave pública e ataques que tentam descobrir alguma informação sobre a mensagem original a partir do texto cifrado são ataques importantes. Esse último define o conceito de segurança semântica e é o mais forte entre os dois. Portanto, essa dissertação analisa as bases de segurança da extensão proposta sobre ambos os ataques.

5.5.1 Recuperação da Chave Privada

A chave privada garante o acesso a todos os dados encriptados. No esquema RSA, as chaves públicas e privadas estão relacionadas através do inverso multiplicativo da aritmética modular, o que torna difícil a recuperação da chave privada a partir da chave pública. O esquema RSA gera a chave privada através da multiplicação de dois números primos grandes, a busca de um número primo entre si com o resultado dessa multiplicação e do cálculo do inverso multiplicativo desse número primo, o que são operações computacionalmente triviais. No entanto, para recuperar essa chave privada tendo a chave pública, precisa-se fatorar a chave pública. Essa operação de fatoração, por sua vez, requer grandes recursos computacionais o que significa que recuperar chaves privadas em chaves RSA de 2048 bits, ou mesmo de 1024 bits, é uma tarefa inviável.

O esquema DGHV gera os elementos da chave pública através de uma sequência de encriptações simétricas do valor zero. Isso significa a escolha de dois números aleatórios q_i e r_i, que são combinados através da expressão $x_i = K_{priv} \times q_i + r_i$. Desse modo, obter a chave privada a partir dos elementos da chave pública requer o cálculo do máximo divisor comum a partir de termos aproximados, o que é um problema difícil de ser resolvido. Esse problema é conhecido como Máximo Divisor Comum Aproximado (*Approximate-GCD*) e Nick Howgrave-Graham [95] analisou-o para dois números. Ele provou a segurança de mecanismos baseados no problema.

Por causa dessa análise, define-se os parâmetros η, γ, ρ, ρ' e τ tanto no esquema DGHV quanto na extensão proposta. Nesses casos, incluir o ruído na chave é a componente fundamental para a segurança. Se não fosse adicionado o ruído, o problema se torna tão simples quando o cálculo do máximo divisor comum (MDC). Portanto, como o ruído requer uma busca exaustiva da chave privada sobre todo o espaço do ruído, quanto maior é o ruído introduzido, maior é a segurança do esquema.

No esquema DGHV original [1], Van Dijk *et al.* adotaram o ruído máximo de $\rho = 2^{\lambda}$, o que implica uma busca exaustiva de $2^{4\lambda}$ para cada par avaliado entre os elementos da chave pública. O procedimento de busca básico precisa descobrir todos os máximos divisores comuns possíveis para cada par de elementos da chave pública. Com esses elementos, define-se um conjunto de possíveis candidatos a chave pública. Esse procedimento deve ser então repetido para todos os pares de elementos da chave pública e o resultado é a interseção dos conjuntos de valores possíveis. Portanto, precisa-se repetir o procedimento até que a interseção seja unitária.

Na extensão proposta, o ruído máximo, ρ, é B^{λ}. Considerando $B = 2$ como a base mínima, o ruído adotado é sempre maior que o ruído adotado no esquema DGHV. De fato, na extensão proposta, o ruído tem um tamanho $log_2\ B$ maior, o que aumenta a complexidade do problema. Por questões de segurança, na proposta apresentada nesta dissertação, optou-se por adotar esse ruído maior. No entanto, acredita-se que uma análise mais formal sobre quanto o aumento do ruído provoca de melhora na segurança do esquema pode mostrar que o aumento do ruído não é necessário. Portanto, mesmo que o aumento de segurança com o aumento do ruído não esteja formalmente analisado nessa dissertação, garante-se que a extensão não reduz a complexidade do problema do MDC Aproximado, pois os fundamentos da redução para o problema são os mesmos.

5.5.2 Segurança Semântica

Um esquema criptográfico semanticamente seguro significa que duas mensagens são completamente indistinguíveis após serem encriptadas com esse esquema. Nessa análise, considera-se um jogo entre um desafiante e um atacante. Inicialmente, o atacante recebe a chave pública do desafiante. Após, o atacante envia duas mensagens diferentes com o mesmo tamanho para o desafiante, que escolhe uma das mensagens aleatoriamente e a encripta. Após receber o texto cifrado, o atacante indica qual mensagem gerou o texto cifrado e vence o jogo se acertar. Dessa forma, o esquema é semanticamente seguro se a probabilidade do atacante vencer o jogo é $1/2 + \epsilon$, para $\epsilon \approx 0$.

No esquema DGHV, a segurança semântica é reduzida para o problema do MDC

Aproximado. Mais especificamente, Van Dijk *et al.* provam que um atacante $\mathfrak{A}$, com vantagem ϵ pode ser convertido para um algoritmo $\mathcal{A}$ para resolver o problema com probabilidade de sucesso de $\epsilon/2$, no mínimo. Uma análise similar pode ser desenvolvida para a proposta de extensão dessa dissertação. No entanto, algumas modificações são necessárias. Primeiro, o tamanho dos parâmetros η, γ e ρ devem ser aumentados por um fator $log_2\ B$, por causa do uso de uma base maior. Além disso, a análise deve considerar o maior espaço de mensagens, o que requer pequenas modificações no procedimento de aprendizado do bit menos significativo dos quocientes das divisões de números aleatório pela chave privada. Em particular, com o objetivo de garantir a indistinguibilidade da distribuição dos textos cifrados, deve-se somar ao parâmetro τ a parcela $log_2\ B$ ao invés de apenas uma unidade.

Na extensão proposta, uma vez que a redução para o problema de MDC aproximado utiliza valores maiores de ruído, a segurança semântica tende a ser maior. No entanto, para avaliar se existe um aumento concreto de segurança no esquema proposto, uma análise mais formal precisa ser feita. Essa análise inclui a redução formal do esquema estendido para o problema do Máximo Divisor Comum Aproximado e não foi realizada devido a sua complexidade e a incerteza sobre os ganhos reais, não representando, assim, uma boa relação custo-benefício. No entanto, apesar da falta da análise formal, observa-se que os níveis de segurança da proposta parecem ser os mesmos que os níveis do esquema DGHV, pois os princípios básicos de ambos os esquemas são mantidos.

5.6 Avaliação Experimental da Proposta

Além da proposta matemática da extensão do esquema DGHV, de Van Dijk *et al.*, esta seção apresenta uma implementação do esquema de criptografia homomórfico de Van Dijk *et al.* e da proposta de extensão do esquema apresentado nesta dissertação. O objetivo da avaliação do sistema é definir a relação entre segurança e utilização de recursos do esquema de encriptação homomórfica. A segurança do sistema é definida pelo parâmetro de segurança (λ) através da relação de proporcionalidade em que quanto maior o valor do parâmetro de segurança, maior é a segurança do esquema. A partir dessa proporcionalidade, toda a avaliação realizada é baseada no parâmetro de segurança (λ).

A Seção 5.6.1 define as métricas utilizadas na avaliação e as expectativas de resultado para cada métrica aplicada. A Seção 5.6.2 apresenta a implementação realizada, bem como suas bibliotecas básicas, desafios encontrados e algoritmos utilizados. Por fim, baseado na implementação desenvolvida, a Seção 5.6.3 apresenta os resultados obtidos através dos testes reais para avaliar o uso de recursos computacionais de processamento e armazenamento do esquema DGHV e da extensão

proposta.

5.6.1 Métricas

Como o foco da avaliação do esquema proposto em comparação com a literatura é verificar a praticidade dos métodos, as métricas utilizadas consideram a carga de processamento envolvida nos algoritmos de Geração de Chaves, Encriptação e Decriptação e a sobrecarga de armazenamento para as chaves e o texto cifrado. A carga de processamento dos algoritmos é calculada pelo tempo de execução de cada um dos algoritmos avaliados. Para verificar a sobrecarga de armazenamento dos esquemas de encriptação propostos, calculou-se o tamanho, em bits, das chaves privada e pública e das cifras geradas pelos algoritmos de encriptação. É fato que o tamanho da chave pública cresce com o aumento do parâmetro de segurança (λ).

De todas as métricas avaliadas, apenas o tamanho da chave privada é determinístico. Para todas as demais métricas avaliadas, existe um fator aleatório na obtenção de seu valor, consideram-se os valores médios, apresentados juntamente com o intervalo de confiança de 95%. Além da variação estatística dos resultados, com as métricas de desempenho, obtêm-se tempos de execução específicos para o cenário de testes utilizados. Portanto, para essas métricas, exibem-se as medidas em valores absolutos e em valores relativos, para que se tenha a noção do crescimento relativo do tempo com o aumento da base B e do parâmetro de segurança, útil para a reprodutibilidade dos resultados, mas sem perder os valores absolutos dos tempos de execução, usados para a análise de praticidade dos esquemas de encriptação.

5.6.2 Implementação

O esquema DGHV de encriptação homomórfica [1] e a extensão proposta foram implementados e analisados quanto ao seu desempenho no tempo de execução e no uso de recursos. A implementação está em Python, com o sorteio dos números realizados através do módulo *random*, da biblioteca padrão e o cálculo de números grandes realizado com as bibliotecas *bigfloat* e *gmpy2*, que se baseiam nas bibliotecas GMP (*The GNU Multiple Precision Arithmetic Library*) e MPFR (*Multiple Precision Floating-Point Reliably*), escritas na linguagem C.

A implementação dos esquemas segue o paradigma de orientação a objetos, com uma classe representando cada esquema homomórfico, além de classes de utilitários para auxiliar nas operações com números grandes. Dado que a proposta de extensão é uma especialização do método original, a classe que representa o esquema de encriptação homomórfico total herda da classe de encriptação homomórfica para utilizar todas as suas operações e parâmetros internos sem a necessidade de re-implementá-los. A Figura 5.1, apresenta o diagrama de classes da implementação,

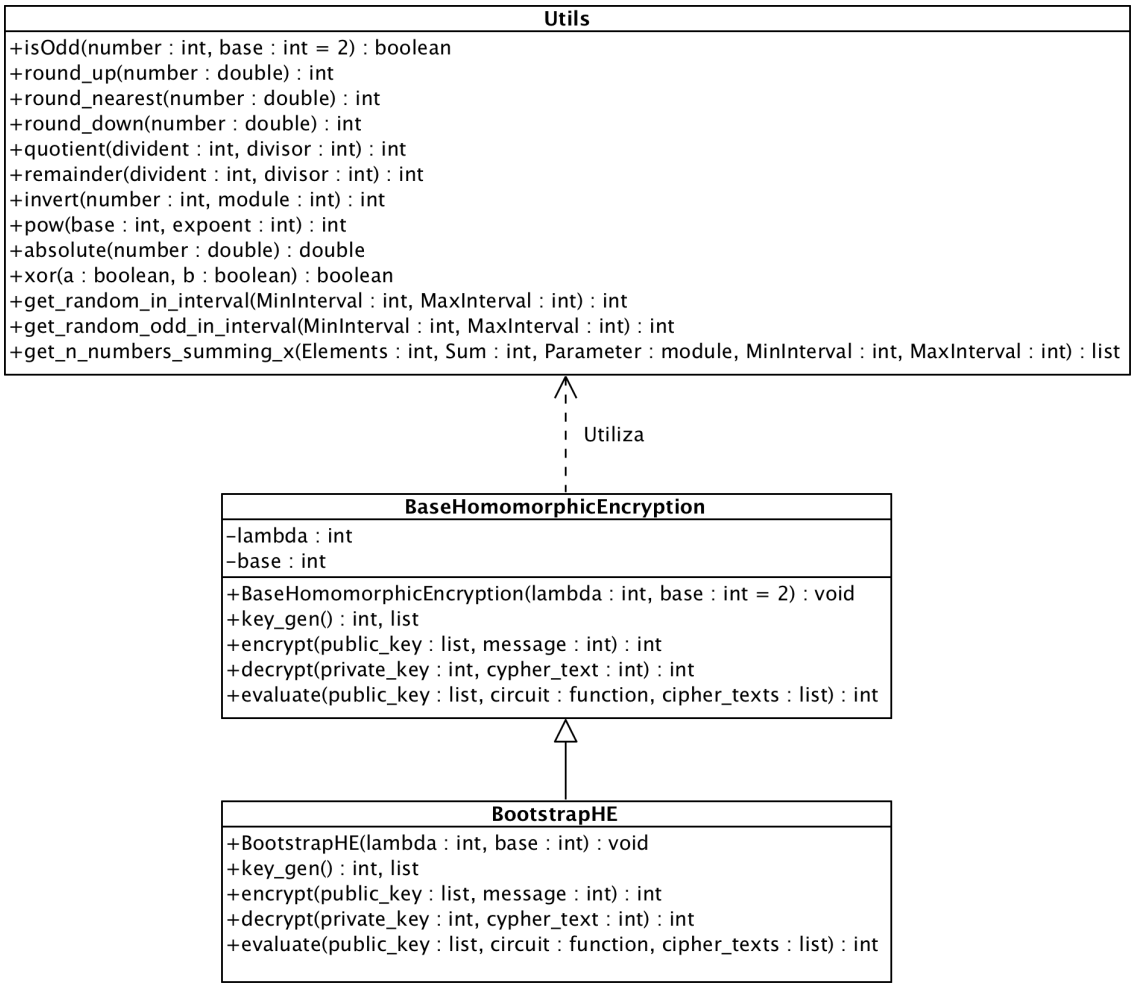


Figura 5.1: Diagrama de classes da implementação.

ilustra essa dinâmica, bem como representa os métodos de geração de chaves, encriptação, decriptação e avaliação, que foram reimplementados na classe do esquema de encriptação homomórfico total.

A estrutura dos métodos é definida juntamente com o esquema de encriptação proposto, definido na Seção 5.1. No entanto, alguns algoritmos, são definidos simplesmente em linguagem natural, mas requerem maior detalhamento de sua implementação. Uma classe com métodos utilitários foi criada para simplificar a implementação do esquema e para permitir a modularização e reutilização de código. Inicialmente, essa seção aborda os métodos utilitários, que são utilizados em ambos os esquemas de encriptação. Os métodos que possuem implementação trivial não são apresentados, mas destaca-se o detalhamento da implementação dos métodos utilitários que não seguem a codificação padrão ou que requerem algum teorema do cálculo numérico.

O primeiro algoritmo a ser detalhado consiste no cálculo da divisão e do resto de números inteiros, que segue a lógica do menor resto, utilizado também pelo esquema

DGHV. A principal diferença entre esse método e o cálculo tradicional de resto de números inteiros está no arredondamento do quociente, que é feito pelo número inteiro mais próximo e não pelo número inteiro inferior. Portanto, na implementação proposta, o resto é o menor possível. Isso implica um ruído menor para o texto cifrado, mas também significa que o algoritmo de Decriptação aceita um valor menor para o ruído, dado que, nesse contexto, a mensagem mais o ruído adicionado deve ser menor do que a metade da chave privada. Caso o método padrão de cálculo do resto fosse utilizado, o ruído poderia ter o tamanho da chave privada.

Dois outros algoritmos conhecidos para aumentar o desempenho das funções padrão foram utilizados, um para exponenciação e outro para cálculo da inversa multiplicativa modular. A operação de exponenciação de g^x requer a realização de x multiplicações. Para reduzir a complexidade da operação, converte-se o expoente x para o número binário $x_b = (x_n x_{n-1} \cdots x_2 x_1 x_0)_2$. Assim, multiplica-se g por g, obtendo g^2, depois g^2 por g^2 obtendo g^4, e assim sucessivamente até obter g^n. Após, multiplica-se as potências de g, no qual x_b, vale 1. O Algoritmo 1 elucida o funcionamento, que pode ser apresentado como o exemplo da exponenciação de g por 54. Toma-se 54 em binário, que é $(110110)_2$, ou seja, $32 + 16 + 4 + 2$. Assim, calcula-se g^2, g^4, g^8, g^{16} e g^{32}. Após, multiplica-se os elementos g^2, g^4, g^{16} e g^{32}, dado que $g^{32+16+4+2} = g^{54}$. Com esse algoritmo, ao invés de realizar 53 multiplicações, realiza-se apenas 8 multiplicações, melhorando o desempenho da computação.

```
   input  : g, x
   output: y = g^x
 1 y ⟵ g;
 2 z ⟵ 1;
 3 x_binary ⟵ ConvertToBinary(x);
 4 for iterator = 0 to n do
 5 |   if x[iterator] == 1 then
 6 |   |   z ⟵ z × y;
 7 |   end
 8 |   y ⟵ y^2;
 9 end
10 return y;
```

Algoritmo 1: Algoritmo para Exponenciação Rápida

O cálculo da inversa multiplicativa modular se baseia no teorema de Fermat generalizado por Euler. A inversa a^{-1} de um número a é definida como o número que, ao ser multiplicado por a, encontra-se 1, e é expressa pela equação

$$a.a^{-1} = 1. \tag{5.25}$$

Em aritmética normal, a inversa de um número inteiro é obrigatoriamente um nú-

mero real entre 0 e 1. No entanto, na aritmética modular de base N, a inversa de um número inteiro é outro número inteiro no intervalo 0 e $N-1$. Outra diferença principal é que, na aritmética modular, nem todos os números possuem inversa. Apenas os números a cujo $mdc(a, N) = 1$ são invertíveis. Assim, denomina-se o conjunto $(\mathbb{Z}_N)^*$ como o conjunto dos números inteiros invertíveis na base N. Assim, em aritmética modular, a inversa a^{-1}, com $a \in (\mathbb{Z}_N)^*$ é dada por

$$a.a^{-1} = 1 \; em \; \mathbb{Z}_N. \tag{5.26}$$

Para calcular a inversa multiplicativa de um número em aritmética modular de forma eficiente, utiliza-se o Algoritmo de Euclides Estendido (AEE). Seja $mdc(a, N) = 1$ e sejam x e y números inteiros tais que, segundo o AEE,

$$a.x + N.y = 1. \tag{5.27}$$

Então, como $N.y$ é múltiplo de N, $a.x = 1(modN)$. Dessa forma, têm-se que $x = a^{-1}$ é a inversa multiplicativa de a na base N.

O cálculo do Algoritmo de Euclides Estendido (AEE) recursivo pode ser observado no Algoritmo 2. Repare que a cada nível de recursividade, o valor de a é reduzido, no mínimo, pela metadde. Dessa forma, o algoritmo possui complexidade $O(log_2 a)$.

```
   input  : a, N
   output: d, x, y
1  if N == 0 then
2  |  return (a, 1, 0);
3
4  (d', x', y') = EuclidesEstendido(N, amodN);
5  (d, x, y) = (d', y', x' − ⌊a/N⌋y');
6  return (d, x, y);
```

Algoritmo 2: Algoritmo para Cálculo do MDC através do Algoritmo de Euclides Estendido.

5.6.3 Resultados

Nesta seção, apresentam-se os resultados práticos da implementação do esquema proposto em comparação com o método original, proposto por Van Dijk *et al.* [1]. Os resultados são divididos em resultados de desempenho e resultados de armazenamento visando agrupar cada tipo de métricas avaliadas. Em todas as métricas, varia-se o parâmetro de segurança (λ), que representa o nível de segurança do esquema de encriptação. Quanto maior o valor de λ, maior é a segurança do sistema,

mas também maiores são os tempos de execução e os tamanhos das chaves e cifras geradas.

Todos os testes da implementação ocorreram em um computador equipado com Intel Xeon X5570 de 2.93GHz, com 96 GB de memória RAM. Os resultados apresentam valores absolutos para o cenário de teste usado e os resultados relativos ao esquema DGHV original. Com o sistema proposto, garantiu-se, também, que toda a computação ocorreu entre memória e processador, evitando-se o uso de partições SWAP, que degradariam significativamente o desempenho do esquema. Os resultados apresentados são valores médios de 100 rodadas e os valores são mostrados com intervalo de confiança de 95%.

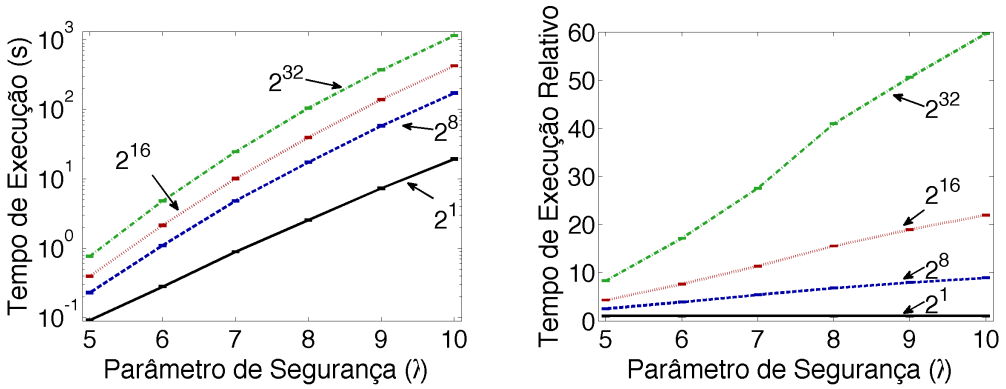
Nos gráficos apresentados, as diversas curvas expressam os diferentes valores para B que foram utilizados na avaliação do esquema proposto. Os valores de B representam a quantidade de números inteiros que podem ser encriptados com o conjunto de chaves gerado e são apresentados como expoentes de base 2 para serem comparados com os tipos básicos de dados numéricos, que são *byte*, *short* e *integer* e possuem, respectivamente, 8, 16 e 32 bits de tamanho. Testes com bases maiores, como 64 bits, também foram desenvolvidos, mas não foi possível tratar a chave pública com o computador utilizado.

A curva $B = 2^1$ possui semântica especial, pois representa o método original proposto por Van Djik et. al. [1]. Nos gráficos com valores relativos, a curva $B = 2^1$ também é usada como unidade de unidade de referência, visto que se pretende avaliar o impacto do da extensão proposta sobre o esquema original. Portanto, nos gráficos relativos, a curva $B = 2^1$ vale 1 para todos os valores do parâmetro de segurança (λ).

Desempenho dos Algoritmos

Nas Figuras 5.2, 5.3 e 5.4, observa-se, respectivamente, uma análise de desempenho baseada no tempo de execução dos algoritmos de Geração de Chaves, Encriptação e Decriptação em função do parâmetro de segurança λ. Os valores avaliados de λ não devem ser usados em aplicações reais devido ao baixo nível segurança. Segundo, Gentry e Halevi [97], $\lambda = 72$ é um valor moderado para o parâmetro de segurança. Os valores baixos λ, no entanto, foram escolhidos para mostrar tendências de desempenho sob as condições computacionais atuais. A maior restrição em se utilizar valores de λ maiores consiste no armazenamento da chave pública, o que atinge tamanhos da ordem de $O(\lambda^{10})$, demandando altas capacidades de memória e armazenamento.

O algoritmo de Geração de Chaves tem o maior custo computacional dentre os três algoritmos básicos de um esquema de encriptação, pois requer a geração de λ^5 elementos da chave pública. No entanto, esse custo não é grave, pois o algoritmo é

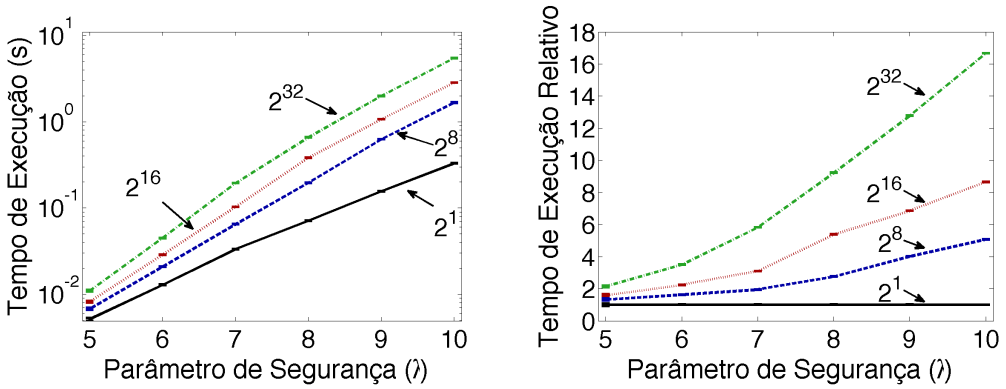


(a) Valores absolutos. (b) Valores em relação a $B = 2^1$.

Figura 5.2: Tempo de execução do algoritmo de Geração de Chaves em função do parâmetro de segurança λ para diferentes valores de B.

executado apenas uma vez para as diversas encriptações, avaliações e decriptações. Através dos resultados apresentados na Figura 5.2, observa-se que a utilização de valores maiores para B implica o aumento do tempo de execução. Esse resultado é proveniente das operações com números maiores, que possuem maior custo de acesso à memória e utilização do processador, pois esses equipamentos trabalham com palavras de tamanho fixo. No entanto, deve-se considerar que, com a proposta dessa dissertação, mais bits são encriptados e processados em cada operação, o que indica melhor desempenho. A Figura 5.2(a) apresenta os tempos de execução do algoritmo de Geração de Chaves em função do parâmetro de segurança λ para diferentes valores de B. Percebe-se, portanto, que o tempo de execução é fortemente impactado com o aumento do parâmetro de segurança λ. No cenário analisado, o esquema DGHV requer em torno de 0,92 segundos para gerar o par de chaves quando $\lambda = 5$, mas, ao aumentar o nível de segurança para $\lambda = 10$, o tempo de execução aumenta para 19,18 segundos aproximadamente, um aumento de mais de 20 vezes. Considerando o parâmetro B, a Figura 5.2(b) mostra a relação entre os tempos de execução de cada base analisada com o esquema DGHV, no qual $B = 2^1$. Nesse contexto, para $\lambda = 10$, que é o caso mais restritivo dentre os analisados, observa-se que uma base capaz de tratar 32 bits por operação tem seu desempenho reduzido em 59,62 vezes. O algoritmo de Geração de chaves é usado apenas uma vez e, posteriormente, o par de chaves pode ser utilizado diversas vezes, ao contrário dos algoritmos de Encriptação e de Decriptação. Assim, o tamanho da mensagem não tem influência nesse algoritmo. Portanto, a extensão proposta reduz o desempenho do algoritmo de Geração de Chaves em relação ao esquema DGHV.

A Figura 5.3 apresenta a análise do tempo de execução para o algoritmo de Encriptação em função o parâmetro de segurança λ para diferentes valores de B. O algoritmo de Encriptação consiste em somar a mensagem original com o ruído do

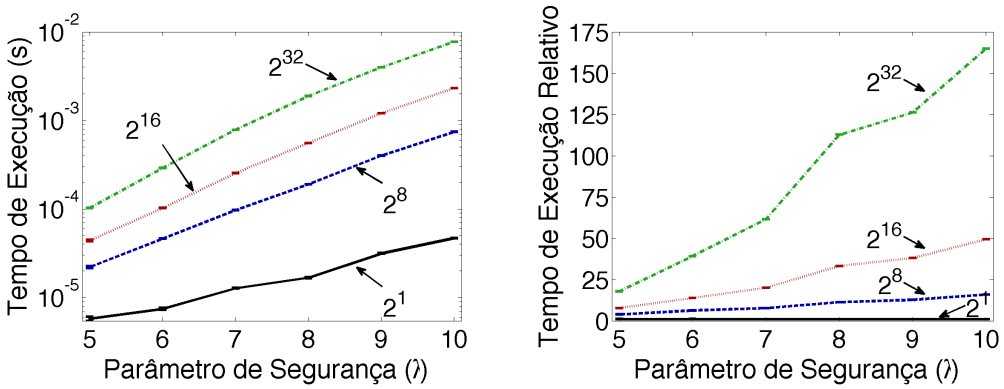


(a) Valores absolutos.

(b) Valores em relação a $B = 2^1$.

Figura 5.3: Tempo de execução do algoritmo de Encriptação em função do parâmetro de segurança λ para diferentes valores de B.

texto cifrado e com os elementos da chave pública escolhidos aleatoriamente. Assim, o principal impacto no desempenho do algoritmo é o aumento do parâmetro de segurança λ, como pode ser observado na Figura 5.3(a). Com os valores de λ analisados, observa-se, para a encriptação de números de 32 bits, o tempo de execução aumentou de 0,1 segundos, com $\lambda = 5$, para 5,4 segundos, com $\lambda = 10$. Esse aumento significativo é proveniente do aumento da chave pública, o que acarreta em um subconjunto aleatório de encriptação maior, em média. Como o algoritmo de Encriptação requer a soma de todos os elementos desse conjunto, quanto maior é o conjunto, maior é o tempo de execução do algoritmo. Contudo, o foco da proposta é no aumento da base B e sobre esse parâmetro, observa-se que o incremento no tempo de execução não é tão grande, como mostrado na Figura 5.3(b). No entanto, embora o aumento da base represente aumentos da ordem de até 16 vezes, com o aumento da quantidade de bits com o aumento da base, pode-se observar um aumento no desempenho em comparação com o esquema DGHV. Por exemplo, verifica-se que, para o nível de segurança $\lambda = 7$ e base $B = 2^{32}$, o tempo de execução do algoritmo é 5,81 vezes maior. Contudo, em uma operação de encriptação na extensão proposta, encriptam-se 32 bits por operação. O esquema original requer que a operação de encriptação seja realizada 32 vezes. Assim, a extensão proposta é 5,51 vezes melhor do que o esquema DGHV original. Para valores menores de λ (como $\lambda = 5$), alcança-se desempenho próximo de 15 vezes melhor. Portanto, o aumento no tempo de execução do algoritmo de Encriptação é menor do que o aumento na quantidade de bits encriptados de uma só vez. Além disso, a figura apresenta um crescimento linear no tempo de execução em relação ao parâmetro B enquanto existe um aumento exponencial no tamanho da mensagem encriptada, indicando melhora de desempenho para a proposta com valores de λ maiores. De fato, vale lembrar, que o aumento do parâmetro de segurança significa um aumento



(a) Valores absolutos.

(b) Valores em relação a $B = 2^1$.

Figura 5.4: Tempo de execução do algoritmo de Decriptação em função do parâmetro de segurança λ para diferentes valores de B.

significativo do tempo de execução, mas esse aumento ocorre para todos os valores de B. O impacto do parâmetro adicionado na proposta não é grave, como observado na Figura 5.3(a).

Ao contrário do algoritmo de Encriptação, para o algoritmo de Decriptação, observa-se piora no desempenho com o aumento da base B, como pode-se observar na Figura 5.4. O simples aumento da base para atender 32 bits representa aumentos de até 2 ordens de grandeza. Ainda assim, na Figura 5.4(a), um aumento significativo com o aumento do parâmetro de segurança λ. Considerando o esquema DGHV, o algoritmo de Decriptação é executado em 0,1 milissegundos para $\lambda = 5$, enquanto leva 7 milissegundos para decriptar um texto cifrado quando $\lambda = 10$. Isso quer dizer que o tempo de execução da decriptação aumenta significativa com o aumento do λ e da base. Como pode ser observado na Figura 5.4(b), o algoritmo de Decriptação piora em desempenho conforme B cresce. Isso ocorre devido a maior complexidade da divisão com a utilização divisores maiores. Como o algoritmo de Decriptação desenvolvido é baseado no resto da divisão pela chave privada K_{priv} e pela base B, a utilização de bases maiores, implica em aumento em ambos os divisores. Por exemplo, para $\lambda = 7$ e $B = 2^{32}$, o tempo de execução é 61,54 vezes maior do para o esquema original. No entanto, considerando os 32 bits decriptados em cada operação, obtém-se uma perda de desempenho total de 1,92 vezes. Observa-se, no entanto, que, embora o tempo execução da decriptação aumente com maior proporção, esse algoritmo executa com um tempo, em média, 3 ordens de grandeza menores do que o algoritmo de Encriptação.

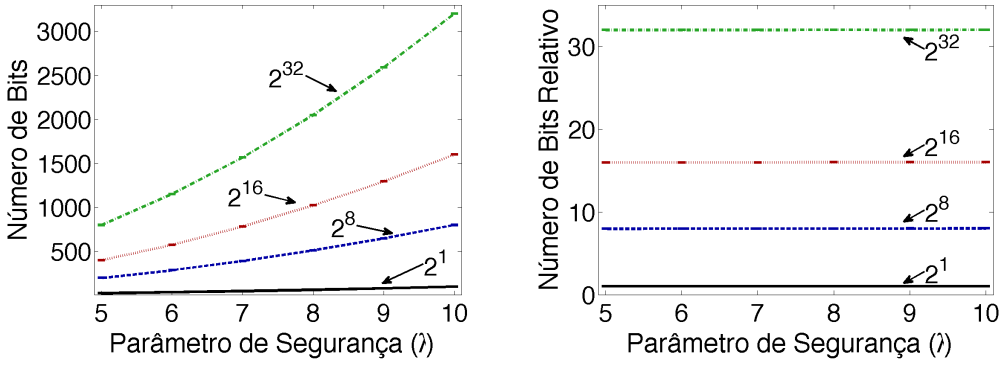
Considerando os ganhos de desempenho com a encriptação e descontando as perdas com a decriptação, a extensão proposta apresenta um ganho de desempenho de 2,87 para o caso analisado, considerando apenas os dados relativos. Quando se avalia o mesmo caso sob a ótica dos valores absolutos, observa-se que o tempo de

execução da encriptação para $B = 2^{32}$ e $\lambda = 7$ é de 193 milissegundos e a decriptação levou 0,78 milissegundos. Para o mesmo valor de λ no esquema DGHV, ou seja, $B = 2^1$, tem-se que a encriptação levou 33 milissegundos e a decriptação foi realizada em 0,01 milissegundos. Contabilizando os totais, tem-se 193,78 milissegundos para $B = 2^{32}$ contra 33,01 milissegundos, o que representa um aumento de 5,87 vezes no tempo de execução global por operação. Contudo, considerando que para processar um número de 32 bits no esquema DGHV precisa-se de 32 operações, o tempo total real no esquema DGHV é de 1.056,32 milissegundos. Assim, a extensão proposta melhora o desempenho dessa operação em 5,45 vezes, mesmo com o maior tempo gasto no algoritmo de Decriptação. Além disso, sabe-se que em um esquema de encriptação homomórfica, o algoritmo de Encriptação é usado mais frequentemente do que o algoritmo de Decriptação, pois deve-se encriptar todos os valores de entrada do circuito computado e a decriptação é realizada apenas do resultado. Se o objetivo fosse armazenar os dados, o algoritmo de Decriptação seria usado mais vezes, pois seria necessário a cada recuperação do dado enquanto o algoritmo de Encriptação seria utilizado apenas no armazenamento. Portanto, como o foco da proposta é o processamento de dados, a extensão proposta melhora consideravelmente o desempenho global de operação.

Armazenamento para as Chaves e Textos Cifrados

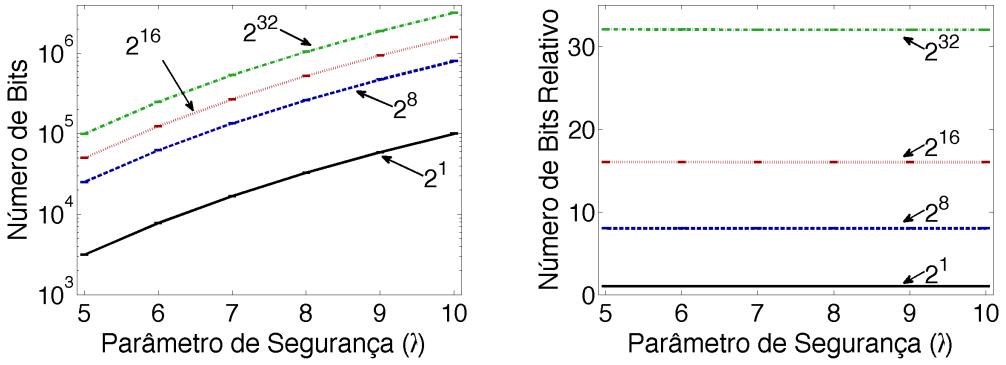
As Figuras 5.5, 5.6 e 5.7 apresentam, respectivamente, uma análise do tamanho da chave privada, chave pública e dos textos cifrados em função do parâmetro de segurança λ para diferentes valores de B. A curva $B = 2^1$ consiste no esquema DGHV, utilizado como referência na análise da extensão proposta. Vale ressaltar ainda, que os valores de λ utilizados não devem ser adotados em ambientes comerciais, pois não oferece nível de segurança aceitável.

A chave privada, K_{priv}, é escolhida no intervalo $\left[2^{\lambda^2-1}, 2^{\lambda^2}\right)$. Assim, K_{priv} tem tamanho determinístico e seu tamanho depende dos valores de λ e B. Quanto maior são λ e B, maior é a chave privada. A Figura 5.5 confirma essa tendência ao comparar o tamanho da chave privada para diferentes valores de B. Nessa avaliação, consideram-se chaves privadas que variam de 25 bits (para $\lambda = 5$) a 3.200 bits (para $\lambda = 10$), como observado na Figura 5.5(a), mas esses valores ainda não são práticos para ambientes de produção não confiáveis. Testes com tamanhos maiores de chave privada são restringidos pelo uso de memória, pois o aumento da chave privada implica grande aumento no tamanho da chave pública, dado que a chave privada é da ordem de $O(\lambda^2 log_2\ B)$ e a chave pública é da ordem de $O(\lambda^{10} log_2\ B)$. Como a Figura 5.5(b) apresenta, o aumento da chave privada com o aumento da base do esquema, observa-se que a chave privada aumenta da ordem de $log_2\ B$. Assim, quanto maior é a quantidade de bits que o par de chaves consegue tratar por



(a) Valores absolutos. (b) Valores em relação a $B = 2^1$.

Figura 5.5: Tamanho da chave privada em função do parâmetro de segurança λ para diferentes valores de B.

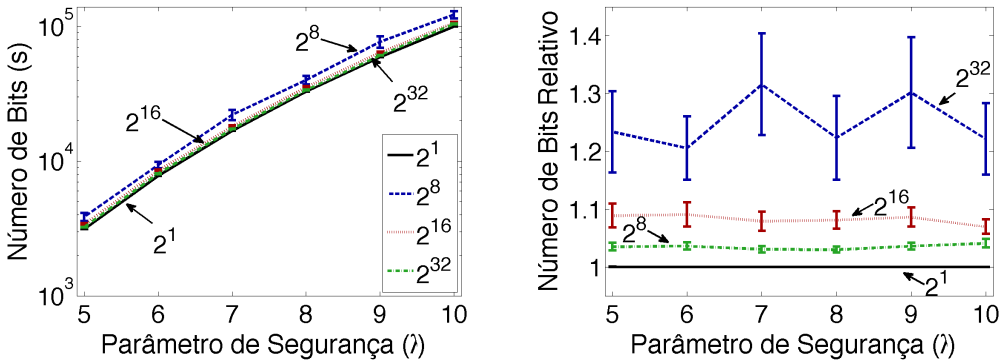


(a) Valores absolutos. (b) Valores em relação a $B = 2^1$.

Figura 5.6: Tamanho do elemento da chave pública em função do parâmetro de segurança λ para diferentes valores de B.

operação, igualmente maior é o tamanho da chave privada. Esse aumento significa que a extensão proposta possui maiores requisitos de armazenamento do que o esquema DGHV, visto que o mesmo par de chaves do esquema DGHV é usado para a encriptação e decriptação de todos os 32 bits da mensagem original.

A chave pública, K_{pub}, cresce conforme o aumento do λ e da B, tanto em número de elementos quanto no tamanho dos elementos que compõe a chave pública. O aumento em número de elementos é determinístico e segue o parâmetro de segurança e a base de acordo com a definição do parâmetro τ, que é dado por $\lambda^5 + \lambda + log_2 B$, como determinado no algoritmo de Geração de Chaves. Assim, quanto maior é o parâmetro de segurança, mais elementos a chave pública tem. Por outro lado, o tamanho dos elementos da chave pública depende do parâmetro q_i, que é escolhido aleatoriamente no intervalo $[0, B^{\lambda^5}/K_{priv})$. Como consequência, avaliar o tamanho dos elementos da chave pública requer uma abordagem estatística. Uma vez que os



(a) Valores absolutos.

(b) Valores em relação a $B = 2^1$.

Figura 5.7: Tamanho do texto cifrado por bit encriptado em função do parâmetro de segurança λ para diferentes valores de B.

elementos da chave pública são gerados pela expressão $x_i = K_{priv} * q_i + r_i$, o valor médio de cada elemento tem magnitude $(B^{\lambda^5})/2$. Assim, o tamanho médio dos elementos da chave pública cresce com o aumento do λ e do B, como confirmado na Figura 5.6. Por exemplo, observa-se que para a encriptação de 32 bits por operação, o algoritmo de Geração de Chaves produz uma chave pública 32 vezes maior. Esse aumento ocorre, pois todos os componentes geradores dos elementos da chave pública (K_{priv}, q_i e r_i) são sorteados em intervalos que dependem do $log_2\ B$. Assim, com $B = 2^{32}$, a chave pública é, em média 32 bits maior do que no esquema DGHV, conforme apresentado na Figura 5.6(b). O principal problema para o armazenamento da chave pública consiste no tamanho total da chave e no aumento do parâmetro λ. Como apresentado na Figura 5.6(a), no esquema DGHV original, enquanto a chave privada para $\lambda = 5$ é 25 bits, cada elemento da chave pública é de 3.120 bits, totalizando 9.768.720 bits (9,7 MBits), considerando os 3.131 elementos presentes nessa chave pública. Dentre os valores de λ avaliados, o tamanho da chave pública aumenta bastante, chegando a mais de 9,3 Gbits quando $\lambda = 10$, em que a chave tem 100.011 elementos, cada um com 99.994 bits, em média. Para chaves que permitem o processamento de 32 bits por operação, o tamanho da chave com $\lambda = 10$ chega a 298,14 Gbits (100.042 elementos com 3,05 Mbits cada). Portanto, como no esquema DGHV um par de chaves é usado para todos os bits a serem encriptados e a extensão proposta requer um par de chaves maior, a proposta utiliza mais recursos de memória e armazenamento.

Finalmente, a Figura 5.7 apresenta o tamanho médio dos textos cifrados gerados pela extensão proposta. Todos os valores são normalizados pelo tamanho da mensagem encriptada com o objetivo de realizar uma comparação mais justa dos requisitos de armazenamento por bit em todas as bases avaliadas. No teste realizado, geraram-se mensagens aleatoriamente no intervalo $[0, B - 1]$. Após a encriptação

da mensagem, divide-se o tamanho do texto cifrado pelo tamanho da mensagem original, ambos em bits. Como apresentado na Figura 5.7(a), o principal elemento para o aumento do texto cifrado é o valor do parâmetro de segurança λ. Com o aumento de λ, o valor de x_0 também aumenta. Como o algoritmo de Encriptação gera o texto cifrado a partir do resto da divisão do valor calculado por x_0, o maior texto cifrado a ser encontrado é $x_0/2$. Dessa forma, o tamanho do texto cifrado acompanha o tamanho do maior elemento da chave pública. Por exemplo, para o esquema DGHV, o tamanho do texto cifrado varia de 3.121 bits, quando $\lambda = 5$, para 99.995 bits, quando $\lambda = 10$. De fato, observa-se que o valor médio do texto cifrado consiste no valor médio dos elementos da chave pública. Na Figura 5.7(b), observa-se que, mesmo com o aumento do tamanho da mensagem, os recursos de armazenamento necessários para encriptar um bit continuam os mesmos. Dessa forma, a extensão proposta provê o mesmo grau de compactação do que o esquema DGHV, embora trabalhe com números inteiros grandes. Observa-se, no entanto, que, para bases menores, como $B = 8$, o grau de compactação é menor, como pode ser visto na Figura 5.7(b) através da curva na qual $B = 2^3$, que é mostrada sobre todas as outras curvas. No entanto, esse aumento na razão entre o tamanho do texto cifrado e o tamanho da mensagem original é menor do que 30%, o que não é crítico, uma vez que o foco principal da proposta é no cálculo de números grandes através de bases grandes. De fato, para bases grandes, o grau de compactação do esquema proposto é semelhante ao grau de compactação do esquema DGHV. Portanto, a extensão proposta, embora trabalhe com números grandes, não piora o grau de compactação do esquema.

Capítulo 6

Conclusão

A construção de uma Internet do Futuro precisa resolver problemas graves da Internet atual, como a segurança, o endereçamento, a mobilidade, a qualidade de serviço oferecida, o roteamento, e muitos outros. Além disso, ser flexível para incorpor as aplicações. Novos modelos de arquitetura de comunicação e de negócios vem sendo propostos para a Internet. Todas as propostas devem ser testadas em escala e tráfego real e a virtualização se mostra como uma técnica apropriada para testar as novas propostas sem interromper a Internet atual.

Na primeira parte dessa dissertação, apresenta-se a ferramenta VNEXT [35], que consiste em uma ferramenta de gerência e controle de plataformas de testes virtualizadas com a plataforma Xen. O VNEXT é resultado de um esforço coletivo do Grupo de Teleinformática e Automação (GTA) e o trabalho desenvolvido e descrito nessa dissertação foi incorporado à ferramenta. As contribuições do mestrado para a ferramenta consistem em um mecanismo de migração ao vivo de roteadores virtuais sem perda de pacotes, o desenvolvimento de uma nova versão do controlador centralizado da ferramenta e a construção de uma interface de visualização da topologia das redes física e virtuais em três dimensões.

O mecanismo de migração presente por padrão na plataforma Xen, utilizada na ferramenta VNEXT foi projetado para a migração de servidores virtuais, uma vez que a plataforma Xen usada foi desenvolvida para consolidação de servidores em *datacenters*. Dessa forma, como os servidores executam suas aplicações sobre o protocolo TCP na maioria das vezes, a perda de alguns pacotes por um período curto de tempo é recuperada pelo mecanismo de recuperação de erros do TCP e, portanto, não é visto como um problema. No entanto, a utilização dessa plataforma Xen para a virtualização de redes implica em algumas mudanças de objetivos. Na camada de rede, por exemplo, não existe nenhum mecanismo de recuperação de perdas de pacotes e, dessa forma, os pacotes perdidos não são recuperados. Procurar evitar as perdas de pacotes no procedimento de encaminhamento é essencial. Dessa forma, para evitar a perda de pacotes, o mecanismo proposto [47] se utiliza da separação

dos planos de controle e de dados no roteador virtual. O plano de controle permanece no roteador virtual. No roteador físico, cria-se uma tabela de encaminhamento própria para o roteador virtual e, com essa tabela, o plano de dados do roteador virtual é atualizado de tempos em tempos de forma a manter a tabela de encaminhamento sincronizada com as informações do plano de controle. Assim, durante todo o processo de migração, o plano de dados está ativo e capaz de encaminhar pacotes, evitando, assim a perda de pacotes durante o período de inatividade do roteador virtual. No entanto, a proposta ainda mantém algumas premissas da migração padrão do Xen que são inadequadas para a migração de roteadores virtuais, como exigir que o disco virtual seja compartilhado entre a máquina física de origem e de destino e que os roteadores físicos tenham os mesmos vizinhos de um salto. Portanto, a proposta melhora o principal problema da migração ao vivo dos roteadores virtuais, que é a perda de pacotes, mas deixa em aberto algumas questões fundamentais para a sua implementação prática.

A ferramenta VNEXT tem o objetivo de ser uma plataforma de testes compartilhada entre diversas universidades e instituições de pesquisa. Nesse cenário, os requisitos de segurança do VNEXT são análogos aos dos sistemas de computação em nuvem e em grade. Precisam-se de mecanismos capazes de garantir a segurança e a privacidade dos dados armazenados e processados. O armazenamento seguro dos dados é um problema largamente estudado e soluções práticas e eficientes são conhecidas e bastante utilizadas. Algoritmos como o 3DES (*Triple Data Encryption Standard*) e o AES (*Advanced Encryption Standard*) possuem reconhecida segurança para manter seguro e privado os dados armazenados em instalações compartilhadas. Por outro lado, com relação ao processamento seguro, não existem soluções práticas e viáveis. Esquemas de criptografia que permitem o processamento das mensagens encriptadas sem precisar descriptografá-las são chamados de esquemas de encriptação homomórfica. O estudo desses esquemas e a proposta de um esquema capaz de realizar operações aritméticas com números inteiros grandes [106] consiste no tema da segunda parte dessa dissertação.

A encriptação homomórfica provê privacidade e segurança para o processamento de dados em ambientes não confiáveis, como a computação em nuvem ou em grade. Gentry [19] mostra que a viabilidade do conceito de encriptação totalmente homomórfica e Van Dijk *et al.* [1] definiu um esquema simples para o conceito. No entanto, ambos os esquemas são limitados para o processamento de um bit por operação. Essa dissertação estendeu o esquema DGHV com o objetivo de permitir operações com números inteiros de tamanho arbitrário sem transformá-los em bits [106]. Provou-se a corretude do esquema proposto através dos requisitos para o tamanho total do ruído agregado no texto cifrado, que deve ser menor do que a metade da chave privada. Assim, quanto menor é a chave privada, menor é o

ruído aceitável e, portanto, menos operações podem ser realizadas. Nesta dissertação, realiza-se uma análise do limite de operações que podem ser realizadas com esquema de encriptação. Conclui-se que a multiplicação é a operação mais crítica e a principal razão para limitar a profundidade do circuito. Por exemplo, considerando $\lambda = 80$ e $B = 2^{32}$, pode-se realizar até 77 multiplicações. Ao mesmo tempo, nessa configuração, pode-se realizar mais de $10^{245.023}$ operações de adição.

O esquema proposto é avaliado considerando os recursos de processamento e o grau de compactação do texto cifrado. Os recursos de processamento são calculados a partir do tempo de execução dos algoritmos. O grau de compactação do texto cifrado de um esquema de encriptação é definido como a razão entre o tamanho do texto cifrado e da mensagem original. Portanto, um esquema compacto possui o mesmo tamanho para o texto cifrado e para a mensagem. Considerando o desempenho, observa-se melhora no processo de encriptação que supera o aumento no tempo de execução do processo de decriptação. Nesse contexto, a extensão proposta apresenta um melhoramento do desempenho global do esquema. Por exemplo, comparando com o esquema DGHV, tempo total compreendido por uma encriptação e uma decriptação é 2,87 vezes melhor para chaves privadas de até 1.568 bits ($B = 2^{32}$ e $\lambda = 7$). Por outro lado, na extensão proposta, armazenar as chaves pública e privada requer mais memória ou espaço em disco porque os intervalos de sorteio dos números aleatórios são aumentados por um fator de $log_2\ B$. Esse aumento, contudo, é benéfico, pois aumenta a segurança do esquema contra ataques de força bruta. Com relação ao problema do Máximo Divisor Comum Aproximado, a extensão proposta mantem os mesmos princípios básicos assumidos pelo esquema DGHV para a redução do esquema para o problema. Portanto, garante-se a inviabilidade de se obter a chave privada a partir da chave pública.

Na avaliação prática, conclui-se que a compactação do esquema é mantida porque o tamanho do texto cifrado por bit é aproximadamente o mesmo, tanto no esquema DGHV quando no proposto. Esse tamanho, contudo, é maior em alguns cenários com bases pequenas, como $B = 8$. No entanto, a extensão proposta não é indicada para cenários com números pequenos porque esses números podem ser facilmente processados em operações que consideram um bit por vez.

Toda a análise realizada sobre a parte da criptografia homomórfica considera o parâmetro λ, que é importate para determinar o nível de segurança do esquema. Conforme λ aumenta, o tamanho das chaves pública e privada, o tamanho do texto cifrado e o tempo de execução aumentam. Desenvolveu-se uma análise matemática para definir os limites de corretude da proposta e, baseado nessa análise, apresentou-se os limites máximos no número de operações de adição ou multiplicação. Observa-se que a extensão proposta é mais eficiente para a encriptação de números inteiros grandes do que o esquema DGHV, pois a proposta consiste na encriptação e no

processamento de mais de um bit por operação. No esquema DGHV, é necessário decompor o número inteiro em bits e, então, encriptar cada um dos bits em uma operação diferente. Além disso, aplicar operações de adição e multiplicação sobre os textos cifrados de mensagens com mais de 1 bit requer funções de adição e multiplicação especiais no esquema DGHV. Por outro lado, com a extensão proposta, utilizam-se as operações padrões para adição e multiplicação, o que significa melhor compatibilidade com programas e linguagens de programação já existentes.

Os resultados experimentais mostram que os requisitos de memória para chaves públicas e privadas são maiores do que os requisitos do esquema DGHV. Mesmo havendo semelhanças entre ambos os esquemas, a extensão proposta requer mais espaço para armazenar o par de chaves conforme o parâmetro B aumenta. Assim, a redução do tamanho das chaves e a melhora da compactação do esquema ainda é um desafio em aberto. Essa melhora é fundamental para a implantação da computação homomórfica em ambientes de não confiáveis, como computação em nuvem e em grades. Outros desafios em aberto na dissertação consistem na análise formal sobre os impactos na segurança do esquema com o aumento da base. Essa análise pode contribuir para a redução de alguns parâmetros para os mesmos utilizados pelo esquema DGHV, o que pode contribuir para a redução do tamanho das chaves e do texto cifrado. Nesse contexto, muito trabalho ainda deve ser feito com o objetivo de reduzir o impacto do crescimento do parâmetro λ para níveis de segurança aceitáveis. Portanto, apesar dos desafios em aberto, conclui-se que a extensão proposta contribui para transformar a encriptação totalmente homomórfica em uma ferramenta prática e viável em ambientes comerciais que requerem processamento em dispositivos não confiáveis.

Referências Bibliográficas

[1] VAN DIJK, M., GENTRY, C., HALEVI, S., et al. "Fully homomorphic encryption over the integers", *Advances in Cryptology–EUROCRYPT 2010*, pp. 24–43, 2010.

[2] VELLOSO, P. B., LAUFER, R. P., DE O CUNHA, D., et al. "Trust management in mobile ad hoc networks using a scalable maturity-based model", *Network and Service Management, IEEE Transactions on*, v. 7, n. 3, pp. 172–185, 2010.

[3] MOREIRA, M., FERNANDES, N., COSTA, L., et al. "Internet do futuro: Um novo horizonte", *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC 2009*, pp. 1–59, 2009.

[4] TAVEIRA, D. M., MORAES, I. M., RUBINSTEIN, M. G., et al. "Técnicas de Defesa Contra Spam". In: *Minicursos do Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais - SBSeg'2006*, pp. 202–250, agosto de 2006.

[5] ABDALLA, M., BRESSON, E., CHEVASSUT, O., et al. "Strong password-based authentication in TLS using the three-party group DiffieHellman protocol", *International Journal of Security and Networks*, v. 2, n. 3, pp. 284–296, 2007.

[6] CLARK, D. D., WROCLAWSKI, J., SOLLINS, K. R., et al. "Tussle in cyberspace: defining tomorrow's Internet", *IEEE/ACM Transactions on Networking*, v. 13, n. 3, pp. 462–475, junho de 2005.

[7] MORAES, I. M., CAMPISTA, M. E. M., MOREIRA, M. D. D., et al. "Distribuição de Vídeo sobre Redes Par-a-Par: Arquiteturas, Mecanismos e Desafios". In: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2008*, pp. 115–171, maio de 2008.

[8] ARMBRUST, M., FOX, A., GRIFFITH, R., et al. "A view of cloud computing", *Communications of the ACM*, v. 53, n. 4, pp. 50–58, 2010.

[9] FELDMANN, A. "Internet clean-slate design: what and why?" *ACM SIGCOMM Computer Communication Review*, v. 37, n. 3, pp. 59–64, julho de 2007.

[10] ALVES, R. S. A., CAMPBELL, I. V., COUTO, R. S., et al. "Redes Veiculares: Princípios, Aplicações e Desafios". In: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2009*, pp. 199–254, Recife, PE, Brasil, maio de 2009.

[11] FERNANDES, N., MOREIRA, M., DUARTE, O. "A self-organized mechanism for thwarting malicious access in ad hoc networks". In: *INFOCOM, 2010 Proceedings IEEE*, pp. 1–5. IEEE, 2010.

[12] VELLOSO, P. B., LAUFER, R. P., DUARTE, O. C., et al. "HIT: A human-inspired trust model". In: *Mobile and Wireless Communication Networks*, Springer, pp. 35–46, 2006.

[13] CLAVEIROLE, T., DE AMORIM, M. D., ABDALLA, M., et al. "Securing wireless sensor networks against aggregator compromises", *Communications Magazine, IEEE*, v. 46, n. 4, pp. 134–141, 2008.

[14] T., O. C., MOREIRA, M. D. D., RUBINSTEIN, M. G., et al. "Redes Tolerantes a Atrasos e Desconexões". In: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2007*, Belém, PA, Brasil, 2007.

[15] FIUCZYNSKI, M. E. "PlanetLab: overview, history, and future directions", *SIGOPS Oper. Syst. Rev.*, v. 40, n. 1, pp. 6–10, 2006.

[16] BARHAM, P., DRAGOVIC, B., FRASER, K., et al. "Xen and the art of virtualization". In: *ACM SIGOPS Operating Systems Review*, v. 37, pp. 164–177. ACM, 2003.

[17] WANG, Y., DER MERWE, J. V., REXFORD, J. "VROOM: Virtual Routers On the Move". In: *Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networking*, pp. 1–7, novembro de 2007.

[18] RIVEST, R., SHAMIR, A., ADLEMAN, L. "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, v. 21, n. 2, pp. 120–126, 1978.

[19] GENTRY, C. "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the 41st annual ACM symposium on Theory of computing*, pp. 169–178. ACM, 2009.

[20] SMART, N., VERCAUTEREN, F. "Fully homomorphic encryption with relatively small key and ciphertext sizes", *Public Key Cryptography–PKC 2010*, pp. 420–443, 2010.

[21] CLARK, D. D., PARTRIDGE, C., RAMMING, J. C., et al. "A knowledge plane for the Internet". In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03)*, pp. 3–10, New York, NY, USA, 2003. ACM. ISBN: 1-58113-735-4.

[22] ANDERSON, T., PETERSON, L., SHENKER, S., et al. "Overcoming the Internet impasse through virtualization", *Computer*, v. 38, n. 4, pp. 34–41, 2005. ISSN: 0018-9162.

[23] FERNANDES, N., MOREIRA, M., MORAES, I., et al. "Virtual networks: Isolation, performance, and trends", *Annals of Telecommunications*, v. 66, n. 5, pp. 339–355, 2011.

[24] FERNANDES, N., DUARTE, O. *VIPER: Fine Control of Resource Sharing in Virtual Networks*. Relatório técnico, UFRJ, 2012.

[25] MOSKOWITZ, R., NIKANDER, P. "Host Identity Protocol (HIP) Architecture". . RFC 4423 (Informational), maio 2006. Disponível em: <`http://www.ietf.org/rfc/rfc4423.txt`>.

[26] LYNO HENRIQUE GONÇALVES FERRAZ. "Uma Avaliação do Protocolo HIP para Provisão de Mobilidade na Internet". In: *Projeto de Graduação*, março de 2010.

[27] DE BRITO, G. M., VELLOSO, P. B., MORAES, I. M. "Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet", *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC*, v. 2012, pp. 211–264, 2012.

[28] RIBEIRO, I. C., GUIMARÃES, F. Q., KAZIENKO, J. F., et al. "Segurança em Redes Centradas em Conteúdo: Vulnerabilidades, Ataques e Contramedidas", *Minicursos do XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2012*, pp. 151–195, 2012.

[29] TORRES, J. V., FERRAZ, L. H. G., DUARTE, O. "Redes orientadas a conteúdo baseadas em controladores hierárquicos", *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos-SBRC*, 2013.

[30] CASE, J., FEDOR, M., SCHOFFSTALL, M., et al. “Simple Network Management Protocol (SNMP)”. . RFC 1157 (Historic), maio 1990. Disponível em: <http://www.ietf.org/rfc/rfc1157.txt>.

[31] LIOTTA, A., PAVLOU, G., KNIGHT, G. “Exploiting agent mobility for large-scale network monitoring”, *Network, IEEE*, v. 16, n. 3, pp. 7–15, 2002.

[32] EGEVANG, K., FRANCIS, P. “The IP Network Address Translator (NAT)”. . RFC 1631 (Informational), maio 1994. Disponível em: <http://www.ietf.org/rfc/rfc1631.txt>. Obsoleted by RFC 3022.

[33] DROMS, R. “Dynamic Host Configuration Protocol”. . RFC 2131 (Draft Standard), mar. 1997. Disponível em: <http://www.ietf.org/rfc/rfc2131.txt>. Updated by RFCs 3396, 4361, 5494.

[34] MOCKAPETRIS, P. “Domain names - implementation and specification”. . RFC 1035 (Standard), nov. 1987. Disponível em: <http://www.ietf.org/rfc/rfc1035.txt>. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966.

[35] PISA, P., COUTO, R., CARVALHO, H., et al. “VNEXT: Virtual network management for Xen-based Testbeds”. In: *Network of the Future (NOF), 2011 International Conference on the*, pp. 41–45. IEEE, 2011.

[36] FEAMSTER, N., GAO, L., REXFORD, J. “How to lease the Internet in your spare time”, *ACM SIGCOMM Computer Communication Review*, v. 37, n. 1, pp. 61–64, janeiro de 2007.

[37] SCHAFFRATH, G., WERLE, C., PAPADIMITRIOU, P., et al. “Network virtualization architecture: Proposal and initial prototype”. In: *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pp. 63–72. ACM, 2009.

[38] ACHEMLAL, M. E. A. *Virtualisation Approach: Concept*. Relatório técnico, The FP7 4WARD Project - Architecture and Design for the Future Internet, 2010.

[39] CHOWDHURY, N., BOUTABA, R. “Network virtualization: state of the art and research challenges”, *Communications Magazine, IEEE*, v. 47, n. 7, pp. 20–26, 2009.

[40] FEAMSTER, N., GAO, L., REXFORD, J. "How to lease the Internet in your spare time", *ACM SIGCOMM Computer Communication Review*, v. 37, n. 1, pp. 61–64, janeiro de 2007.

[41] CHOWDHURY, N., RAHMAN, M., BOUTABA, R. "Virtual network embedding with coordinated node and link mapping". In: *INFOCOM 2009, IEEE*, pp. 783–791. IEEE, 2009.

[42] FAJJARI, I., AITSAADI, N., PUJOLLE, G., et al. "VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic". In: *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–6. IEEE, 2011.

[43] CARVALHO, H., DUARTE, O. "VOLTAIC: volume optimization layer to assign cloud resources". In: *Proceedings of the 3rd International Conference on Information and Communication Systems*, p. 3. ACM, 2012.

[44] MATTOS, D., MAURICIO, L., CARDOSO, L., et al. "Uma Rede de Testes Interuniversitaria com Técnicas de Virtualizaç ao Hıbridas". In: *Salão de Ferramentas do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 8, maio de 2012.

[45] MCKEOWN, N., ANDERSON, T., BALAKRISHNAN, H., et al. "OpenFlow: enabling innovation in campus networks." *Computer Communication Review*, v. 38, n. 2, pp. 69–74, 2008.

[46] CHOWDHURY, N., ZAHEER, F., BOUTABA, R. "iMark: An identity management framework for network virtualization environment". In: *Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on*, pp. 335–342. IEEE, 2009.

[47] PISA, P., FERNANDES, N., CARVALHO, H., et al. "OpenFlow and Xen-Based Virtual Network Migration", *Communications: Wireless in Developing Countries and Networks of the Future*, pp. 170–181, 2010.

[48] NG, W., JUN, D., CHOW, H., et al. "MIBlets: A practical approach to virtual network management". In: *Integrated Network Management, 1999. Distributed Management for the Networked Millennium. Proceedings of the Sixth IFIP/IEEE International Symposium on*, pp. 201–215. IEEE, 1999.

[49] FERNANDES, N., DUARTE, O. "XNetMon: A network monitor for securing virtual networks". In: *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–5. IEEE, 2011.

[50] NIEBERT, N., BAUCKE, S., EL-KHAYAT, I., et al. "The way 4WARD to the creation of a future Internet". In: *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pp. 1–5. IEEE, 2008.

[51] EGI, N., GREENHALGH, A., HANDLEY, M., et al. "Forwarding path architectures for multicore software routers". In: *Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow*. ACM, 2010.

[52] EGI, N., GREENHALGH, A., HANDLEY, M., et al. "Evaluating xen for router virtualization". In: *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pp. 1256–1261. IEEE, 2007.

[53] VAUGHAN-NICHOLS, S. "New approach to virtualization is a lightweight", *Computer*, v. 39, n. 11, pp. 12–14, 2006.

[54] SZEGEDI, P., FIGUEROLA, S., CAMPANELLA, M., et al. "With evolution for revolution: Managing federica for future internet research", *Communications Magazine, IEEE*, v. 47, n. 7, pp. 34–39, 2009.

[55] HE, J., ZHANG-SHEN, R., LI, Y., et al. "Davinci: Dynamically adaptive virtual networks for a customized internet". In: *Proceedings of the 2008 ACM CONEXT Conference*, p. 15. ACM, 2008.

[56] LXCENTER. "HyperVM - OpenVz, Xen, Windows Virtualization Manager". . Disponível em: <`http://lxcenter.org/software/hypervm`>.

[57] BEGNUM, K. "Managing large networks of virtual machines". In: *Proceedings of the 20th Large Installation System Administration Conference*, pp. 205–214, 2006.

[58] FRAGNI, C., MACIEL KOSMALSKI COSTA, L. "ECO-ALOC: Energy-Efficient Resource Allocation for Cluster-Based Software Routers", *Computer Networks*, 2012.

[59] DOS SANTOS ALVES, R., CAMPISTA, M., COSTA, L. "Um Servidor de Máquinas Virtuais Adaptado a Multiplas Pilhas de Protocolos". In: *XVI Workshop de Gerência e Operação de Redes e Serviços (WGRS 2011) - SBRC'2011*. Citeseer, maio de 2011.

[60] ALVES, R. D. S., CAMPISTA, M. E. M., COSTA, L. H. M. K., et al. "Towards a pluralist internet using a virtual machine server for network customization". In: *Proceedings of the Asian Internet Engineeering Conference*, AINTEC '12, pp. 9–16, New York, NY, USA, 2012. ACM.

[61] COUTO, R., CAMPISTA, M., COSTA, L. "XTC: a throughput control mechanism for Xen-based virtualized software routers". In: *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pp. 1–6. IEEE, 2011.

[62] BOURGUIBA, M., HADDADOU, K., PUJOLLE, G. "Evaluating Xen-based virtual routers performance", *International Journal of Communication Networks and Distributed Systems*, v. 6, n. 3, pp. 268–282, 2011.

[63] BERL, A., RACE, N., ISHMAEL, J., et al. "Network virtualization in energy-efficient office environments", *Computer Networks*, v. 54, n. 16, pp. 2856–2868, 2010.

[64] SHERWOOD, R., GIBB, G., YAP, K., et al. "Flowvisor: A network virtualization layer", *OpenFlow Switch Consortium, Tech. Rep*, 2009.

[65] EGI, N., GREENHALGH, A., HANDLEY, M., et al. "Evaluating Xen for Router Virtualization". In: *ICCCN'07: International Conference on Computer Communications and Networks*, pp. 1256–1261, agosto de 2007.

[66] CLARK, C., FRASER, K., HAND, S., et al. "Live migration of virtual machines". In: *NSDI'05: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, pp. 273–286, Berkeley, CA, USA, 2005. USENIX Association.

[67] WANG, Y., KELLER, E., BISKEBORN, B., et al. "Virtual Routers on the Move: Live Router Migration as a Network-Management Primitive". In: *ACM SIGCOMM*, pp. 231–242, agosto de 2008.

[68] RIVEST, R., ADLEMAN, L., DERTOUZOS, M. "On data banks and privacy homomorphisms", *Foundations of secure computation*, pp. 169–179, 1978.

[69] MORAIS, E., DAHAB, R. "Encriptação homomórfica", *Minicursos do XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2012*, pp. 151–195, 2012.

[70] VAIKUNTANATHAN, V. "Computing Blindfolded: New Developments in Fully Homomorphic Encryption". In: *Proceedings of the 52nd Annual*

Symposium on Foundations of Computer Science, pp. 5–16. Springer-Verlag, 2011.

[71] CHOWN, P. "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)". . RFC 3268 (Proposed Standard), jun. 2002. Disponível em: <`http://www.ietf.org/rfc/rfc3268.txt`>. Obsoleted by RFC 5246.

[72] GOLDWASSER, S., MICALI, S. "Probabilistic encryption", *Journal of computer and system sciences*, v. 28, n. 2, pp. 270–299, 1984.

[73] SHANNON, C. "Communication theory of secrecy systems", *Bell system technical journal*, v. 28, n. 4, pp. 656–715, 1949.

[74] KATZ, J., LINDELL, Y. *Introduction to modern cryptography*. London and New York, CRC Press, 2008.

[75] BONEH, D., HALEVI, S., HAMBURG, M., et al. "Circular-secure encryption from decision diffie-hellman". In: *Advances in Cryptology–CRYPTO 2008*, Springer, pp. 108–125, 2008.

[76] ELGAMAL, T. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *Advances in Cryptology*, pp. 10–18. Springer, 1985.

[77] PAILLIER, P. "Public-key cryptosystems based on composite degree residuosity classes". In: *Advances in Cryptology—EUROCRYPT'99*, pp. 223–238. Springer, 1999.

[78] DAMGÅRD, I., JURIK, M. "A generalisation, a simplification and some applications of paillier's probabilistic public-key system". In: *Public Key Cryptography*, pp. 119–136. Springer, 2001.

[79] AJTAI, M., DWORK, C. "A public-key cryptosystem with worst-case/average-case equivalence". In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 284–293. ACM, 1997.

[80] LEVIEIL, E., NACCACHE, D. "Cryptographic test correction", *Public Key Cryptography–PKC 2008*, pp. 85–100, 2008.

[81] REGEV, O. "New lattice-based cryptographic constructions", *Journal of the ACM (JACM)*, v. 51, n. 6, pp. 899–942, 2004.

[82] BONEH, D., GOH, E., NISSIM, K. "Evaluating 2-DNF formulas on ciphertexts", *Theory of Cryptography*, pp. 325–341, 2005.

[83] LÓPEZ, J., DAHAB, R. *An Overview of Elliptic Curve Cryptography*. Relatório técnico, Unicamp, 2000.

[84] GENTRY, C., HALEVI, S., VAIKUNTANATHAN, V. "A simple BGN-type cryptosystem from LWE", *Advances in Cryptology–EUROCRYPT 2010*, pp. 506–522, 2010.

[85] SANDER, T., YOUNG, A., YUNG, M. "Non-interactive cryptocomputing for NC^1". In: *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pp. 554–566. IEEE, 1999.

[86] MELCHOR, C., GABORIT, P., HERRANZ, J. "Additively homomorphic encryption with d-operand multiplications", *Advances in Cryptology–CRYPTO 2010*, pp. 138–154, 2010.

[87] FELLOWS, M., KOBLITZ, N. "Combinatorial cryptosystems galore!" *Contemporary Mathematics*, v. 168, pp. 51–51, 1994.

[88] COHEN, J., FISCHER, M. "A robust and verifiable cryptographically secure election scheme". In: *26th Annual Symposium on Foundations of Computer Science*, pp. 372–382. IEEE, 1985.

[89] ADIDA, B. "Helios: Web-based open-audit voting". In: *USENIX Security Symposium*, v. 17, pp. 335–348. USENIX Association, 2008.

[90] CHOR, B., GOLDREICH, O., KUSHILEVITZ, E., et al. "Private information retrieval". In: *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pp. 41–50. IEEE, 1995.

[91] KUSHILEVITZ, E., OSTROVSKY, R. "Replication is not needed: Single database, computationally-private information retrieval". In: *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pp. 364–373. IEEE, 1997.

[92] ISHAI, Y., PASKIN, A. "Evaluating branching programs on encrypted data", *Theory of Cryptography*, pp. 575–594, 2007.

[93] HÅSTAD, J., IMPAGLIAZZO, R., LEVIN, L. A., et al. "A pseudorandom generator from any one-way function", *SIAM Journal on Computing*, v. 28, n. 4, pp. 1364–1396, 1999.

[94] CORON, J., MANDAL, A., NACCACHE, D., et al. "Fully homomorphic encryption over the integers with shorter public keys", *Advances in Cryptology–CRYPTO 2011*, pp. 487–504, 2011.

[95] HOWGRAVE-GRAHAM, N. "Approximate integer common divisors", *Cryptography and Lattices*, pp. 51–66, 2001.

[96] CORON, J., NACCACHE, D., TIBOUCHI, M. *Optimization of Fully Homomorphic Encryption.* Relatório técnico, Cryptology ePrint Archive, Report 2011/440, 2011., 2011.

[97] GENTRY, C., HALEVI, S. "Implementing Gentry's fully-homomorphic encryption scheme", *Advances in Cryptology–EUROCRYPT 2011*, pp. 129–148, 2011.

[98] BRAKERSKI, Z., VAIKUNTANATHAN, V. "Efficient fully homomorphic encryption from (standard) LWE". In: *FOCS 2011*, pp. 97–106, 2011.

[99] SILVA, R., DE A CAMPELLO, A., DAHAB, R. "LWE-based identification schemes". In: *Information Theory Workshop (ITW), 2011 IEEE*, pp. 292–296, 2011.

[100] BRAKERSKI, Z., GENTRY, C., VAIKUNTANATHAN, V. "Fully homomorphic encryption without bootstrapping", *ITCS 2012*, 2012.

[101] LYUBASHEVSKY, V., PEIKERT, C., REGEV, O. "On ideal lattices and learning with errors over rings", *Advances in Cryptology–EUROCRYPT 2010*, pp. 1–23, 2010.

[102] SMART, N., VERCAUTEREN, F. "Fully homomorphic SIMD operations", *Designs, Codes and Cryptography*, pp. 1–25, 2011.

[103] BRAKERSKI, Z., VAIKUNTANATHAN, V. "Fully homomorphic encryption from ring-LWE and security for key dependent messages", *Advances in Cryptology–CRYPTO 2011*, pp. 505–524, 2011.

[104] NAEHRIG, M., LAUTER, K., VAIKUNTANATHAN, V. "Can homomorphic encryption be practical?" In: *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pp. 113–124. ACM, 2011.

[105] BONEH, D., SEGEV, G., WATERS, B. "Targeted malleability: Homomorphic encryption for restricted computations". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 350–366. ACM, 2012.

[106] PISA, P. S., ABDALLA, M., DUARTE, O. C. M. B. "Somewhat Homomorphic Encryption Scheme for Arithmetic Operations on Large Integers". In: *4th Global Information Infrastructure and Networking Symposium - GIIS-2012*, p. 8p, Choroni, Venezuela, dezembro de 2012. IEEE.