MINISTÉRIO DA DEFESA

EXÉRCITO BRASILEIRO

DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA

INSTITUTO MILITAR DE ENGENHARIA

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

EDWARD CÉSPEDES CARAGEORGE

RENAN GEMIGNANI

AVALIAÇÃO DA SEGURANÇA DA CRIPTOGRAFIA BASEADA EM RETICULADO

RIO DE JANEIRO

2013

INSTITUTO MILITAR DE ENGENHARIA

EDWARD CÉSPEDES CARAGEORGE

RENAN GEMIGNANI

AVALIAÇÃO DA SEGURANÇA DA CRIPTOGRAFIA BASEADA EM RETICULADO

Iniciação à Pesquisa apresentada ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia.

Orientador: José Antônio Moreira Xexéo – D.Sc..

RIO DE JANEIRO

2013

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

621.39 Carageorge, Edward Céspedes
C258a Avaliação da Segurança de Criptografia Baseada em Reticulado / Edward Céspedes Carageorge, Renan Gemignani; orientado por José Antônio Moreira Xexéo - Rio de Janeiro : Instituto Militar de Engenharia, 2013.

48 p.: il.

Iniciação à Pesquisa (IP) - Instituto Militar de Engenharia - Rio de Janeiro, 2013.

1. Engenharia de Computação. 2. Criptografia. 3. Avaliação de Segurança.
I. Gemignani, Renan II. Xexéo, José Antônio Moreira III. Título IV. Instituto Militar de Engenharia.

CDD 621.39

INSTITUTO MILITAR DE ENGENHARIA

EDWARD CÉSPEDES CARAGEORGE

RENAN GEMIGNANI

AVALIAÇÃO DA SEGURANÇA DA CRIPTOGRAFIA BASEADA EM RETICULADO

Relatório de iniciação à pesquisa apresentado ao Curso de Engenharia de Computação do Instituto Militar (IME) de Engenharia, como parte das exigências do IME.

Aprovado em 04/06/2013 pela seguinte Banca Examinadora:

Orientador: Prof. José Antônio Moreira XEXÉO – D.Sc., do IME

ANDERSON Fernandes P. dos Santos, D.Sc., do IME

Julio Cesar DUARTE, D.Sc., do IME

RIO DE JANEIRO

2013

“ A primeira coisa a entender é que você não entende”.

SOREN AABYE KIERKEGAARD

SUMÁRIO

SUMÁRIO 6

LISTA DE ILUSTRAÇÕES 8

1 INTRODUÇÃO 12
1.1 MOTIVAÇÃO 12
1.2 OBJETIVO 12
1.3 JUSTIFICATIVA 13
1.4 METODOLOGIA 13
1.5 ESTRUTURA 14

2 CONCEITOS BÁSICOS DE CRIPTOGRAFIA 15
2.1 ENCRIPTAÇÃO E DECRIPTAÇÃO 15
2.1.1 FUNCIONAMENTO BÁSICO DE UM CRIPTOSSISTEMA 15
2.1.2 CRIPTOGRAFIA SIMÉTRICA 15
2.1.3 CRIPTOGRAFIA ASSIMÉTRICA 16
2.1.4 FUNÇÕES DE MÃO ÚNICA COM ARAPUCA 16
2.2 COMPUTAÇÃO QUÂNTICA E CRIPTOGRAFIA PÓS-QUÂNTICA 17

3 CONCEITOS BÁSICOS DE RETICULADOS 19
3.1 INTRODUÇÃO 19
3.2 DEFINIÇÕES E PROPRIEDADES 19
3.3 PROBLEMAS DIFÍCEIS EM RETICULADOS 21
3.4 ALGORITMO DE BABAI 22

4 CRIPTOSISTEMAS BASEADOS EM RETICULADOS 24
4.1 GGH 24
4.2 AJTAI-DWORK 26
4.3 NTRU 27
4.3.1 RETICULADO NTRU 29
4.4 LWE 30
4.4.1 O PROBLEMA LWE 30

5 CRIPTOANÁLISE 32
5.1 REDUÇÃO GAUSSIANA 32
5.2 REDUÇÃO LLL 33
5.3 KZ 34
5.4 BKZ 37
5.5 BKZ 2.0 38

6 AVALIAÇÃO DE SEGURANÇA 40
6.1 GERAÇÃO DE RETICULADOS 40
6.2 COMO MEDIR A SEGURANÇA 41
6.3 SEGURANÇA DO LWE E SIS 42

7 CONCLUSÃO 44

8 REFERÊNCIAS 46

LISTA DE ILUSTRAÇÕES

FIG. 3.2.1 Representação gráfica de um reticulado bidimensional 20

FIG. 3.2.2 Duas bases de um mesmo reticulado 21

LISTA DE SIGLAS

apprCVP	approach CVP
apprSVP	approach SVP
CVP	Closest Vector Problem
GGH	Goldreich, Goldwasser e Halevi
LLL	Lenstra, Lenstra e Lovász
LWE	Learning with errors
SIS	Smallest Integer Solution
SVP	Shortest Vector Problem

RESUMO

Criptossistemas baseados em reticulados são candidatos promissores para uma era de criptografia pós-quântica, em que os principais criptossistemas do mundo (RSA e Diffie-Hellman) seriam facilmente quebrados. Embora haja resultados significativos no avanço do desenvolvimento da criptografia baseada em reticulado, o estudo do seu nível de segurança em relação a ataques de redução ainda está em aberto. Neste trabalho, apresentaremos alguns criptossistemas baseados em reticulados, métodos de redução de reticulados e analisaremos como os criptossistemas se comportam frente aos ataques de redução. O trabalho se baseia em criptossistemas do tipo aprendendo com erros, do inglês *learning with errors* (LWE) e nos permitirá comparar a eficiência e segurança de diferentes esquemas criptográficos.

Palavras-chave: Criptografia, reticulado, avaliação de segurança, *learning with errors*.

ABSTRACT

Cryptosystems based on lattices are promising candidates for an era of post-quantum cryptography, in which the main cryptosystems used nowadays would be easily broken. Though there have been significant results in the advancement of lattice-based cryptography, their security with respect to lattice reduction attacks is still an open problem. Here, we present some lattice-based cryptosystems and lattice reduction methods, and we analyze how the cryptosystems behave when subjected to a lattice reduction attack. The studies here are based mostly on the learning with errors (LWE) problem, and will allow us to compare efficacy and security of other cryptosystems as well.

Keywords: Cryptography, lattice, security evaluation, learning with errors.

1 INTRODUÇÃO

O interesse pelo estudo da criptografia é tão antigo quanto a própria escrita. Em uma era onde muita informação trafega por meio de canais de transmissão cuja segurança não pode ser assegurada, como é a internet hoje, o estudo de métodos por meio dos quais a informação que trafega não possa ser utilizada por um intruso mesmo que este tenha acesso ao canal torna-se crítico de modo a evitar que as informações captadas tenham seu sigilo violado. Dentro do estudo da criptografia, há uma preocupação constante com a manutenção da segurança das informações transmitidas mesmo quando sujeitas a um ataque futuro por um ator, que teria acesso a ferramentas com poder computacional exponencialmente maior do que o existente hoje. Uma destas ferramentas que pode vir a ser utilizada num futuro próximo, a computação quântica, pode por em xeque a segurança de criptossistemas que atualmente são largamente utilizados, o que faz com que o estudo de criptossistemas resistentes a este tipo de ataque seja de notável importância.

1.1 MOTIVAÇÃO

A criptografia baseada em reticulados vem recebido certa atenção nos últimos 20 anos, quando alguns pesquisadores conseguiram realizar avaliações de pior caso para alguns criptossistemas [10]. Esta avaliação fora primeiramente apresentada por Ajtai [9], que provou que poder resolver uma instância aleatória de um criptossistema em reticulado implica ser capaz de resolver todas as instâncias do problema, tornando a análise de complexidade média igual à complexidade de pior caso. Contudo, os métodos de ataque a criptossistemas de reticulados ainda não foram estudados com maturidade matemática suficiente de modo a garantir a possibilidade de sucesso em um ataque a um texto cifrado aleatório, o que realça a importância de serem realizados estudos neste âmbito.

1.2 OBJETIVO

Este trabalho possui como objetivo geral o estudo de alguns criptossistemas baseados em reticulados e de algoritmos de redução de reticulados, além de realizar uma avaliação acerca da segurança dos criptossistemas estudados ante os ataques vistos.

Como objetivos específicos, temos o estudo dos problemas difíceis GGH, NTRU, LWE (*learning with errors* - aprendendo com erros, em tradução livre) e SIS (*small integer solution* - pequena solução inteira, em tradução livre), o estudo de criptossistemas baseados nestes problemas e dos métodos de redução de reticulado LLL, KZ e BKZ, bem como definir algumas diretrizes que podem ser utilizadas para avaliar a segurança dos criptossistemas apresentados contra um ataque de redução por um dos métodos estudados.

1.3 JUSTIFICATIVA

Com o advento da computação quântica, e de algoritmos quânticos capazes de decriptar mensagens cifradas com os criptossistemas mais utilizados atualmente (como o RSA e o problema de logaritmos discretos) [3], faz-se a necessidade da criação de novos criptossistemas que levem em conta a possibilidade de ataque por um algoritmo quântico. Postula-se que os problemas baseados em reticulados são imunes a ataques quânticos, o que os torna um interessante objeto de estudo para um possível futuro no qual computadores quânticos são mais acessíveis. Contudo, esta maior resistência a ataques quânticos tem como contrapartida o maior tamanho das chaves envolvidas - de complexidade $\Omega\left(n^2 log\left(n\right)\right)$ dado um parâmetro natural do sistema n - em relação ao dos sistemas supracitados. Felizmente, podem ser usados certos reticulados ideais na maior parte dos casos, o que reduz o tamanho da chave $\Omega\left(n.log\left(n\right)\right)$ [11]. Busca-se utilizar chaves tão pequenas quanto possível, de modo a diminuir o processamento necessário para a realização das operações de encriptação e decriptação sem, contudo, prejudicar a segurança e a integridade do criptossistema ante um ataque. Assim, procura-se desenvolver criptossistemas que são tanto viáveis para utilização de um ponto de vista computacional quanto seguros ante um ataque.

1.4 METODOLOGIA

Primeiramente, foram estudados os criptossistemas GGH e NTRU, servindo como introdução ao assunto devido à sua maior simplicidade em relação aos outros problemas difíceis estudados, bem como os artigos descrevendo o método de redução LLL, utilizado como base para o desenvolvimento dos outros métodos de redução aqui vistos. Então, passou-se ao estudo dos problemas e métodos de redução mais complexos. Com o entendimento dos métodos apresentados, foram procuradas referências de análises da segurança dos criptossistemas ante cada método de ataque visto.

1.5 ESTRUTURA

O trabalho está estruturado da seguinte forma: na seção 2, são apresentados conceitos básicos de criptografia necessários para o entendimento do estudo dos criptossistemas e métodos de redução. Na seção 3, é introduzido o conceito de reticulado e são apresentados de modo geral os problemas sobre os quais os criptossistemas estudados são baseados. Na seção 4 e 5, são descritos os criptossistemas propriamente ditos e os métodos principais de ataque aos criptossistemas. Na seção 6, culminam as conclusões retiradas da bibliografia a respeito da segurança dos criptossistemas ante um possível ataque, e na seção subsequente, são apresentadas as referências do trabalho.

2 CONCEITOS BÁSICOS DE CRIPTOGRAFIA

2.1 ENCRIPTAÇÃO E DECRIPTAÇÃO

2.1.1 FUNCIONAMENTO BÁSICO DE UM CRIPTOSSISTEMA

Conforme [5], um criptossistema (ou cifra) consiste em um meio de se transmitir uma mensagem, denominada texto pleno, de tal modo que ela somente pode ser compreendida pelo receptor desejado. Este processo se dá em duas partes: em primeiro lugar, o emissor encripta a mensagem, transformando-a em outro texto, denominado texto cifrado. Este texto cifrado é transmitido de tal forma que o receptor original possa realizar o processo inverso (denominado decriptação), obtendo deste modo o texto pleno novamente. A este processo, chama-se criptografia.

A encriptação e a decriptação normalmente se dão por meio de um algoritmo que depende de uma outra informação, uma chave, para fazer a encriptação ou a decriptação. Assim, se a mensagem for interceptada por alguém que não possua a chave, esta entidade não poderia recuperar o texto pleno, garantindo assim a segurança.

2.1.2 CRIPTOGRAFIA SIMÉTRICA

A criptografia simétrica é uma forma de criptografia em que a codificação e a decodificação da mensagem são realizadas utilizando a mesma chave pelo emissor e pelo receptor. Esse tipo de técnica transforma um texto claro em texto cifrado utilizando uma chave secreta e um algoritmo de criptografia. Usando a mesma chave e um algoritmo de decriptação, o texto claro é recuperado a partir do cifrado.

2.1.3 CRIPTOGRAFIA ASSIMÉTRICA

O maior problema em relação à segurança de criptossistemas simétricos é o problema da troca de chaves. Uma vez que não há garantias de que a chave não será usurpada por terceiros, um interceptor que conseguisse a chave poderia decriptar e encriptar mensagens utilizando a mesma, quebrando a segurança do sistema. Uma alternativa é a utilização da criptografia assimétrica, em que existem duas chaves relacionadas - um par de chaves. Uma chave pública é disponibilizada livremente para qualquer pessoa que queira enviar uma mensagem. Uma segunda chave, chamada privada é mantida em segredo, para que somente o receptor da mensagem a conheça.

Mensagens criptografadas usando a chave pública só podem ser decriptadas, aplicando um outro algoritmo usando a chave privada correspondente.

Isso significa que não existe a preocupação de transmitir chaves públicas em canais inseguros, como por exemplo na Internet, uma vez que a posse da mesma não permite decriptar mensagens destinadas ao receptor. Em contrapartida, algoritmos de criptografia assimétrica normalmente executam a encriptação e a decriptação em complexidades maiores do que algoritmos de criptografia simétrica, exigindo maior poder de processamento. Como exemplo disso, para uma mensagem de tamanho n o one-time pad (exemplo clássico de criptossistema simétrico) encripta e decripta a mensagem em tempo $O(n)$, enquanto o RSA (um dos criptossistemas assimétricos mais estudados) realiza os mesmos em complexidade $O(n^2)$ - considerando n o número de bits da mensagem.

2.1.4 FUNÇÕES DE MÃO ÚNICA COM ARAPUCA

Uma vez apresentado o funcionamento básico desejado de um criptossistema assimétrico, há a necessidade de gerar criptossistemas propriamente ditos. Um dos meios mais utilizados para se construir criptossistemas assimétricos é utilizando uma função simples de se calcular, mas difícil (em termos computacionais) de ser invertida.

Definição 2.1. Uma função f é dita *de mão única* se valem:

1. Existe um algoritmo de complexidade polinomial no tempo que, dado $x \in Dom(f)$, calcula $f(x)$;

2. Não existe um algoritmo de complexidade polinomial no tempo (no caso médio) tal que, dado apenas y, encontre um elemento da pré-imagem de y (algum $x \in Dom(f)$ tal que $f(x) = y$).

Para algumas funções de mão única, há uma certa informação cujo conhecimento torna possível a resolução do problema delineado acima. A esta informação denomina-se arapuca.

Definição 2.2. Se uma função f é de mão única, pode-se encriptar uma mensagem $x \in Dom(f)$ usando $f(m)$ como o texto cifrado. Como f é de mão única, a busca pelo texto claro m torna-se inviável para grandes valores de m, a não ser que se conheça a arapuca. Deste modo, a mensagem só pode ser decriptada eficientemente por um receptor que possua a arapuca.

2.2 COMPUTAÇÃO QUÂNTICA E CRIPTOGRAFIA PÓS-QUÂNTICA

Com o progresso de estudos na área de mecânica quântica, foi-se notado que alguns fenômenos quânticos não pederiam ser simulados de modo eficiente em uma máquina de Turing[26] (ou seja, em qualquer computador comum). Contudo, um computador que utilizasse os próprios fenômenos quânticos para descrever uma distribuição de estados em apenas um bit (que passa a ter a denominação de qubit) de informação poderia ser utilizado de modo a realizar cálculos com complexidades muito menores. Deste modo, alguns problemas que seriam considerados difíceis no paradigma computacional da mecânica clássica teriam sua solução em tempo razoável tornada possível. Já existem computadores quânticos rudimentares, e, dado tempo suficiente, sua construção em maior escala pode ser tida como inevitável.

A possibilidade do advento de computadores quânticos comerciais é uma ameaça concreta aos sistemas criptográficos baseados na fatoração de inteiros e no problema do logaritmo discreto, uma vez que para ambos os problemas já existem algoritmos quânticos que os resolvem em tempo polinomial [3]. Em resposta a isso, surgiu o campo de estudo da criptografia pós-quântica, que estuda algoritmos criptográficos que seriam resistentes a uma criptoanálise quântica. Como exemplos de classes de algoritmos atualmente pertencentes a este campo de estudo, podemos mencionar os baseados em *hash* e os em Reticulados. Teoriza-se que estes

algoritmos demorariam um tempo exponencial para serem quebrados, mesmo em computadores quânticos.

3 CONCEITOS BÁSICOS DE RETICULADOS

3.1 INTRODUÇÃO

Um reticulado é um subgrupo discreto do $\mathbb{R}^n$ gerado por uma base de vetores, formando um conjunto de pontos em um arranjo de rede. Um reticulado pode ser visto como um análogo discreto de um espaço vetorial, diferindo deste devido ao fato de as combinações lineares dos vetores da base pertencentes ao reticulado serem apenas aquelas com coeficientes inteiros.

3.2 DEFINIÇÕES E PROPRIEDADES

Conforme apresentado em [4].

Definição 3.1. Sejam $v_1, ..., v_n \in \mathbb{R}^m$ um conjunto de vetores linearmente independentes (LI). O reticulado $\mathcal{L}$ gerado por $v_1, ..., v_n$ é dado por:

$$\mathcal{L} := \{\sum_{i=1}^{n} a_i v_i;\ a_1, a_2 ... a_n \in \mathbb{Z}\} \tag{3.1}$$

Podemos representar os vetores $w_1, ..., w_n \in \mathcal{L}$ em termos de $v_1, ...v_n$ com uma transformação linear:

$$w_i = \sum_{j=1}^{n} a_{ij} v_j \tag{3.2}$$

Agora, para representarmos $v_1, ..., v_n$ em termos de $w_1, ..., w_n$, precisamos inverter a matriz A dada por:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

a fim de obtermos:

$$v_i = \sum_{j=1}^{n} a_{ij}^{-1} w_j \tag{3.3}$$

A FIG. 3.2.1 representa o arranjo dos pontos de $\mathcal{L}$ para o caso bidimensional ($m = 2$).

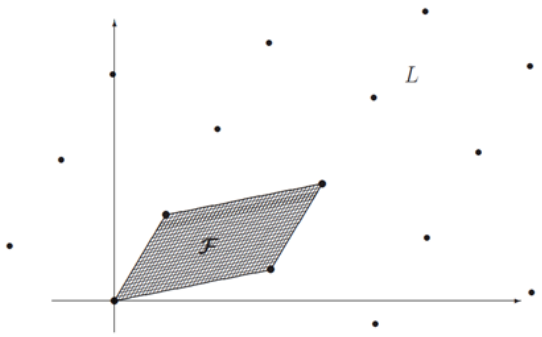

FIG. 3.2.1: Representação gráfica de um reticulado bidimensional

Definimos também a taxa de Hadamard de uma base $V = (v_1, ..., v_n)$ como a quantidade:

$$H(V) = \left(\frac{|det(L)|}{||v_1||.||v_2||...||v_n||} \right)^{\frac{1}{n}} \tag{3.4}$$

Em que L é qualquer base do reticulado, uma vez que todas tem o mesmo determinante[5]. Pelos teoremas de Minkowski e Hermite, $0 \leq H(B) \leq 1$. Quanto mais próxima de 1 for a taxa de Hadamard, mais ortogonais e pequenos, em módulo, serão os vetores geradores do reticulado. Estas bases são chamadas de 'boas'. Caso contrário, ou seja, vetores com tamanho muito grande e pouco ortogonais entre si, elas são ditas 'ruins'.

Dois casos especiais acontecem quando o valor da taxa de Hadamard assume os valores 0 ou 1. Para 0, significa que os vetores são linearmente dependentes e, para esse caso em especial, é irrelevante o tamanho dos vetores. Para a taxa igual a 1, temos vetores completamente ortogonais entre si, e também, não depende do tamanho dos vetores dessa base.

Essas bases são exemplificadas na FIG. 3.2.2 em um exemplo bidimensional.

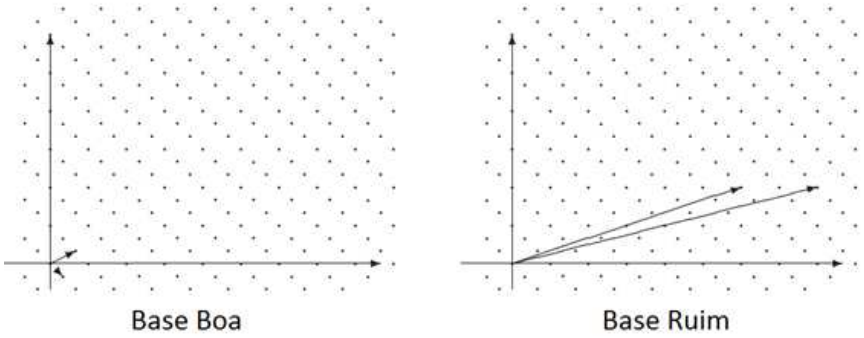

FIG. 3.2.2: Duas bases de um mesmo reticulado

3.3 PROBLEMAS DIFÍCEIS EM RETICULADOS

Nesta seção, citam-se alguns dos principais problemas difíceis usados pela criptografia baseada em reticulado.

1. O Problema do Menor Vetor (SVP): encontrar um vetor não nulo v em um reticulado $\mathcal{L}$ que minimize a norma euclidiana de $||v||$.

2. O Problema do Menor Vetor Aproximado (SVPγ): dado um fator de aproximação $\gamma \geq 1$ encontrar um vetor não nulo v em um reticulado $\mathcal{L}$ tal que $0 \leq ||v|| \leq \gamma$.SVP.

3. O Problema do Vetor mais Próximo (CVP): dado um vetor $w \in \mathbb{R}^m/\mathcal{L}$, encontrar um vetor $v \in \mathcal{L}$ que minimize a norma euclidiana de $||v - w||$.

4. O Problema do Vetor mais Próximo Aproximado (CVPγ): dado um vetor $w \in \mathbb{R}^m/\mathcal{L}$ e um fator de aproximação $\gamma \geq 1$, encontrar um vetor $v \in \mathcal{L}$ tal que $||v - w|| \leq \gamma.\text{CVP}(w, v)$.

   Os próximos problemas difíceis (SIS e LWE) são chamados problemas modulares. Um reticulado $\mathcal{L} \subset \mathbb{Z}^m$ é chamado modular com módulo q ou q-ary, se $q\mathbb{Z}^m \subset \mathcal{L}$. Tais reticulados são interessantes se $q \ll vol(\mathcal{L})$. Usaremos reticulados da forma $\mathcal{L}_{A,q} = \{x \in \mathbb{Z}^m | Ax \equiv 0 \ (mod\ q)\}$, em que A é uma matriz $n \times n$ de coeficientes inteiros congruos a q.

5. Solução do Menor Inteiro (SIS) [16]: dado um módulo q, uma matriz A $(mod\ q)$ e um $w < q$, encontrar $v \in \mathbb{Z}^m$ tal que $Av \equiv 0(mod\ q)$ e $||v|| \leq w$.

   Embora o problema não esteja formulado como um problema em reticulado, é fácil ver que este é um problema no reticulado $\mathcal{L}_{A,q}$.

6. Aprendendo com erros (LWE) [17]: dado um módulo q. Para $s \in \mathbb{Z}_q^n$ e uma distribuição de probabilidade $\mathcal{X}$ em $\mathbb{Z}_q$, seja $A_{s,\mathcal{X}}$ a distribuição de probabilidade em $\mathbb{Z}_q^n \times \mathbb{Z}_q$com amostragem da forma: dado $a \in \mathbb{Z}_q^n$ uniforme, $e \in \mathbb{Z}_q$ de acordo com $\mathcal{X}$, calcule $(a, \langle a, s\rangle + e)\ (mod\ q)$. O problema LWE se resume a dados n, q, $\mathcal{X}$ e um número de amostras independentes de $A_{s,\mathcal{X}}$, encontrar s.

3.4 ALGORITMO DE BABAI

Se um reticulado $\mathcal{L} \subset \mathbb{R}^n$ tem uma base $V = (v, ...v_n)$ consistindo de vetores ortogonais dois a dois, ou seja, $v_i.v = 0,\ \forall i \neq j$ é fácil resolver os problemas SVP e CVP.

Para resolver o SVP, basta observar que o comprimento de qualquer vetor em $\mathcal{L}$ é dado pela fórmula:

$$||a_1v_1 + ... + a_nv_n||^2 = a_1^2||v_1||^2 + ... + a_n^2||v_n||^2 \tag{3.5}$$

Sendo que $a_1, a_2...a_n \in \mathbb{Z}$, a solução do SVP estará no conjunto 3.6

$$\{\pm v_1, \pm v_2, ..., \pm v_n\} \tag{3.6}$$

Analogamente, para encontrarmos o vetor em $\mathcal{L}$ mais próximo de w, ou seja, resolver o CVP, primeiro, fazemos $w = \sum_{i=1}^n b_iv_i$, e $u = \sum_{i=1}^n a_iv_i$.

Sendo que $a_1, a_2...a_n \in \mathbb{Z}$, u será a solução do CVP quando $a_i = [b_i]$, onde $[\cdot]$ denota o inteiro mais próximo.

Em termos simples, se os vetores da base forem razoavelmente ortogonais uns aos outros, então se consegue provável sucesso ao resolver o CVP. No entanto, se os vetores da base forem muito pouco ortogonais e grandes, o algoritmo não funcionará muito bem, e dará um resultado errado.

Em resumo, o Algoritmo de Babai pode ser escrito como:

Algoritmo 1. 1. *Sendo* $V = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$ *uma base boa, escreva* $w_V = wV^{-1}$;

2. *Calcule* $w_V^{'} = [w_V]$;

3. *Retorne* $u = w_V^{'} V$.

Exemplo. Seja $\mathcal{L} \subset \mathbb{R}^2$ um reticulado de base $v_1 = (137, 312)$ e $v_2 = (215, -187)$ e $w = (53172, 81743)$ tal que $w \notin \mathcal{L}$. Usando o algoritmo de Babai, procura-se o vetor $v \in \mathcal{L}$ que é mais próximo de w.

$w = t_1 v_1 + t_2 v_2$

$\begin{cases} 137t_1 + 275t_2 = 53172 & t_1 \approx 296,85 \\ 312t_1 - 187t_2 = 81743 & t_2 \approx 58,15 \end{cases}$

$v = [t_1]\, v_1 + [t_2]\, v_2 = (53159, 81818)$

$||v - w|| \approx 76,12$

Por este resultado, observa-se que v é próximo de w. Este resultado se deve ao fato de que a os vetores da base v_1 e v_2 que geram o reticulado $\mathcal{L}$ possuem uma taxa de Hadamard muito próximo de 1.

$H(v_1, v_2) = \left(\frac{det(L)}{||v_1||.||v_2||} \right) \approx 0,977$

Exemplo. Agora, usando-se uma outra base $v_1' = (1975, 438)$ e $v_2' = (7548, 1627)$ que gere o mesmo reticulado $\mathcal{L}$ e possua taxa de Hadamard próxima de 0.

$H(v_1, v_2) = \left(\frac{det(L)}{||v_1'||.||v_2'||} \right) \approx 0,077$

$w = t_1' v_1' + t_2' v_2'$

$\begin{cases} 1975t_1' + 438t_2' = 53172 & t_1' \approx 5722,66 \\ 7542t_1' + 1627t_2' = 81743 & t_2' \approx -1490,34 \end{cases}$

$v' = [t_1']\, v_1' + [t_2']\, v_2' = (56405, 82444)$

$||v' - w|| \approx 3308,12$

Pelo resultado, conforme era esperado, v' não é tão próximo de w quanto v, do exemplo anterior.

4 CRIPTOSISTEMAS BASEADOS EM RETICULADOS

4.1 GGH

O criptosistema GGH, apresentado por Goldreich, Goldwasser e Halevi [6], é baseado na dificuldade de se encontrar uma solução correta para o CVP.

Para codificar uma mensagem, precisa-se escolher uma base boa (V) - cuja taxa de Hadamard é proxima de um. Esta base servirá como chave privada e a partir dela criar uma base ruim (W), usada como chave pública. Para criação de W, faz-se uso de uma matriz inteira U de determinante com módulo 1 (de maneira que o produto UV também seja uma base do reticulado), cujos coeficientes são grandes, de forma a garantir que a taxa de Hadamard para W seja muito próxima de zero.

A geração das chaves para o reticulado pode ser feita da seguinte maneira: primeiro uma base ortogonal ou "quase ortogonal" é gerada. Pode-se obter tal base escolhendo aleatoriamente coeficientes inteiros, com o cuidado de verificar se os vetores são linearmente independentes, até que os vetores escolhidos satisfaçam à condição da taxa de Hadamard.

Em seguida a base pública é gerada a partir da base privada. Encontrada uma matriz U com determinante de módulo 1, multiplica-se U pela matriz cujas colunas são os vetores da base V.

A mensagem codificada e será o produto entre o texto original m, que é um vetor, e a chave pública. No entanto, para evitar um ataque de texto escolhido [7], faz-se uso de um pequeno vetor r (de tamanho menor do que o menor dentre os vetores da base V), normalmente um Hash do texto original.

Algoritmo 2 (Esquema de Encriptação do GGH [5])**.**

CRIAÇÃO DA CHAVE

Escolher uma base boa $V = \{v_1, \ldots, v_n\}$;
Escolher uma matriz inteira U tal que $det\ U = \pm 1$;
Calcular a base ruim $W = w_1, \ldots, w_n$, em que $W = UV$;
Chave pública: W;
Chave privada: V.

ENCRIPTAÇÃO

Escolher um vetor plaintext $m = m_1, m_2, ..., m_n$;
Escolher um vetor pequeno aleatório r, normalmente um hash de m;
Calcular o texto cifrado $e = m_1 w_1 + ... + m_n w_n + r$.

DECRIPTAÇÃO

Usar o algoritmo de Babai para calcular o vetor $v \in$
$\mathcal{L}$ mais próximo de e;
Calcular vW^{-1} para recuperar m.

O GGH é um tipo de criptografia probabilística, pois um único texto leva a diversos textos cifrados pela simples mudança do fator aleatório (r). Por isso, para que o criptossistema seja eficaz no momento da decodificação, é necessário que a perturbação seja determinada pela aplicação de uma função *hash* ao texto original (m).

Exemplo. Ilustra-se o GGH com um exemplo tridimensional

CRIAÇÃO DA CHAVE

Chave privada: $V = \{v_1, v_2, v_3\}$

$V = \{v_1,\ v_2,\ v_3\} =$
$\{(-97, 19, 19)\ ,\ (-36, 30, 86)\ ,\ (-184, -64, 78)\}$
$H(V) \approx 0,0000208$

Matriz inteira: $U = \begin{pmatrix} 4327 & -15447 & 23454 \\ 3297 & -11770 & 17871 \\ 5464 & -19506 & 29617 \end{pmatrix}$

Chave pública: $W = \{w_1,\ w_2,\ w_3\}$

$w_1 = (-4179163,\ -1882253,\ 583183)$
$w_1 = (-3184353,\ -1434201,\ 444361)$
$w_1 = (-5277320,\ -2376852,\ 736426)$

ENCRIPTAÇÃO

Texto claro $m = \{86, -35, -32\}$
Vetor pequeno aleatório $r = \{-4, -3, 2\}$
Texto cifrado $e = m_1 w_1 + ... + m_n w_n + r = (-79081427, -35617462, 11035473)$

DECRIPTAÇÃO

Usar o algoritmo de Babai para calcular o vetor $v \in \mathcal{L}$ mais próximo de e

$$e \approx 81878,97v_1 - 292300,00v_2 + 443815,04v_3$$
$$v = 81879v_1 - 292300v_2 + 443815v_3$$

Calcular vW^{-1} para recuperar m.

$$v = 86w_1 - 35w_2 - 32w_3$$
$$m = \{86, -35, -32\}$$

$\|e - v\| \approx 5,3852$

ATAQUE MAL SUCEDIDO

Usar o algoritmo de Babai para calcular o vetor $v \in \mathcal{L}(w_1, w_2)$ mais próximo de e

$$e \approx 75,76w_1 - 34,52w_2 - 24,18w_3$$
$$v' = 76w_1 - 35w_2 - 24w_3$$

Calcular vW^{-1} para recuperar m.

$$v = (-79508353, -35809745, 11095049)$$

$\|e - v'\| \approx 472000$

4.2 AJTAI-DWORK

Miklós Ajtai e Cynthia Dwork [8] propuseram um criptossistema baseado em reticulados, usando uma variante do SVP. Infelizmente o tamanho das chaves públicas no criptossistema Ajtai-Dwork é $O(n^4)$ e cada bit de texto claro é expandido para $O(n^2)$ bits encriptados. Nguyen demonstrou que qualquer versão eficiente do criptossistema seria insegura. Não entraremos nos detalhes deste criptossistema.

O criptossistema NTRU, proposto por Hoffstein, Pipher e Silverman [12], édescrito usando anéis polinomiais, mas pode também ser interpretado como sendo o problema de resolver o SVP ou CVP em uma classe especial de reticulados.

O sistema já foi quebrado em alguns casos, no entanto para certas escolhas de parâmetros, ainda é seguro. Essas escolhas serão discutidas em capítulos posteriores.

Antes de apresentarmos o esquema de encriptação do NTRU, vamos fazer algumas cosiderações a respeito de anéis polinomiais e das notações usadas.

$R = \frac{\mathbb{Z}[x]}{(x^N-1)}$,$R_p = \frac{(\mathbb{Z}/p\mathbb{Z})[x]}{(x^N-1)}$,$R_q = \frac{(\mathbb{Z}/q\mathbb{Z})[x]}{(x^N-1)}$

A teoria necessária para o entendimento dos anéis polinomiais é apresentada no apêndice desse relatório.

Seja $a(x) \in R_q$. O center lift de a é um polinômio único $a' \in R\,(x)$ satisfazendo:

$$a'(x) \equiv a(x)\ mod\ q$$

É fácil ver que a transformação $a(x) \to a'(x)$ é reversível, fato este que será importante mais à frente na decriptação de mensagens.

cujos coeficientes estão no intervalo: $-q/2 < a'_i < q/2$

$$T(d_1, d_2) = \left\{ a(x) \in \mathbb{R} : \begin{array}{l} a(x) \text{ possui } d_1 \text{ coeficientes iguais a 1.} \\ a(x) \text{ possui } d_2 \text{ coeficientes iguais a -1.} \\ a(x) \text{ possui todos os outros coeficientes iguais a 0.} \end{array} \right\}$$

O criptosistema do NTRU baseia-se na dificuldade de resolver o seguinte problema em espaços polinomiais discretos: Dado um polinômio $h(x)$, encontrar $f(x) \in T(d+1, d)$, $g(x) \in T(d, d)$ tais que $f \star h \equiv g\ (mod\ q)$. Esse problema é quase-certamente equivalente ao SVP em uma classe de reticulados descita na próxima seção.

Algoritmo 3 (Esquema de encriptação do NTRU [5])**.**

CRIAÇÃO DOS PARÂMETROS PÚBLICOS

Escolher os parâmetros (N, p, q, d)

N e p são primos

$mdc(p, q) = mdc(N, q) = 1$

$q > (6d + 1)p$

CRIAÇÃO DA CHAVE

Escolher um polinômio $f \in T(d+1, d)$.
Escolher um polinômio $g \in T(d, d)$.
Calcular F_q e F_p, inversos de f em R_q e R_p.
Chave Privada: (f, g, F_p, F_q)
Chave pública: $h = F_q * g$.

ENCRIPTAÇÃO

Escolher um vetor plaintext $m \in$
R_p e coeficientes m_i tais que $-p/2 < m_i < p/2$.
Escolher um polinômio aleatório $r \in T(d, d)$.
Calcular o texto cifrado $e \equiv pr * h + m \pmod{q}$.

DECRIPTAÇÃO

Calcular $f * e \equiv pg * r + f * m \pmod{q}$.
Calcula-se o centerlift de a modulo q.
Calcular $m' \equiv F_p * a' \pmod{p}$.
A mensagem original será o center lift de m' modulo p.

Exemplo. Um exemplo numérico para o NTRU

CRIAÇÃO DOS PARÂMETROS PÚBLICOS

Escolher os parâmetros $(N, p, q, d) = (7, 3, 41, 2)$

CRIAÇÃO DA CHAVE

$f(x) = x^6 - x^4 + x^3 + x^2 - 1$
$g(x) = x^6 + x^4 - x^2 - x$
$F_q(x) = 8x^6 + 26x^5 + 31x^4 + 21x^3 + 40x^2 + 2x + 37$
$F_p(x) = x^6 + 2x^5 + x^3 + x^2 + x + 1$
Chave Privada: $(f,\ g,\ F_p,\ F_q)$
Chave pública: $h(x) = 20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30$

ENCRIPTAÇÃO

Texto claro: $m(x) = -x^5 + x^3 + x^2 - x + 1$
Polinômio aleatório: $r(x) = x^6 - x^5 + x - 1$
Texto cifrado: $e(x) \equiv 31x^6 + 19x^5 + 4x^4 + 2x^3 + 40x^2 + 3x + 25 \pmod{q}$

DECRIPTAÇÃO

Calcular $f * e \equiv x^6 + 10x^5 + 33x^4 + 40x^3 + 40x^2 + x + 40 \pmod{q}$
Centerlift: $a(x) = x^6 + 10x^5 - 8x^4 - x^3 - x^2 + x - 1 \pmod{q}$
Calcular $m' \equiv 2x^5 + x^3 + x^2 + 2x + 1 \pmod{p}$
$m(x) = -x^5 + x^3 + x^2 - x + 1$

4.3.1 RETICULADO NTRU

Pode-se utilizar uma classe especial de reticulados para formular o NTRU como sendo a solução de um SVP. Esta versão modificada do problema também foi apresentada por Hoffstein, Pipher e Silverman [12].

Se $h(x) = \sum h_i x^i$ for uma chave pública de NTRU e $h = \begin{pmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}$ é a matriz de convolução de h em R, defina o reticulado associado $R_h^{NTRU} = \begin{pmatrix} I_n & h \\ 0 & qI_n \end{pmatrix}$, em que I_n é a matriz identidade de ordem n.

O vetor resultante da concatenação de f e g (f, g) pertence ao reticulado gerado pela matriz R_h^{NTRU}.

Também podemos ver outros detalhes: dados (N, p, q, d) parâmetros do NTRU.

- $det(R_h^{NTRU}) = q^N$
- $\|(f, g)\| \approx SVP$

Esta última observação vem da heurística de Gauss para o cálculo do tamanho aproximado das soluções do SVP[5].

Assim, com grande probabilidade, (f, g) e suas rotações (uma vez que todas elas pertencem ao reticulado) são as soluções do SVP no reticulado R_h^{NTRU}. Isso mostra uma equivalência entre a formulação do NTRU em anéis polinomiais e o SVP.

4.4 LWE

Em 2005, Regev [13] descreveu um criptossistema baseado em reticulados e no problema *learning with errors*, brevemente apresentado em seções anteriores.

4.4.1 O PROBLEMA LWE

O problema LWE é baseado em um experimento aleatório, baseado na dificuldade de se diferenciar uma função com uma componente aleatória (definida como sendo uma perturbação que obedece a uma distribuição de probabilidade χ de um distribuição uniforme sobre $\mathbb{Z}_q$. Sejam, $n, m, q \in \mathbb{Z}$ são inteiros e $\mathcal{X}$ é uma distribuição de probabilidades sobre $\mathbb{Z}_q$.

O experimento é descrito a seguir:

1. $A \in_R \mathbb{Z}_q^{m \times n}$
2. $v_0 \in_R A$
3. $s \in_R \mathbb{Z}_q^n$ e $\epsilon_{\mathcal{X}} \mathbb{Z}_q^m$, $v_1 = As + e$
4. $b \in_R \{0, 1\}$
5. Envie v_b para receptor
6. Receptor envia bit b' de volta
7. O resultado do experimento é 1 se e somente se $b = b'$

O adversário obtém sucesso no experimento LWE quando consegue distinguir $As + e$ de $v \in_R \mathbb{Z}_q^m$

Definição 4.1 (Problema LWE)**.** O problema LWE consiste em obter probabilidade um no experimento LWE.

Acredita-se que o problema LWE é difícil [13] (e portanto que somente adversários rodando em tempo exponencial em n possam obter probabilidade um). Com o problema em mãos, pode-se criar um criptossistema baseado nele.

Algoritmo 4 (Esquema de encriptação do LWE [1])**.**

CRIAÇÃO DOS PARÂMETROS PÚBLICOS

Escolher m vetores $a_1, a_2, ...a_m \in \mathbb{Z}_q^n$ uniformemente.
Escolher, também aleatoriamente, erros $e_1, e_2...e_m \in \mathbb{R}/\mathbb{Z} \mod 1$

CRIAÇÃO DA CHAVE

Escolher um $s \in \mathbb{Z}_q^n$ aleatoriamente, que servirá como chave privada.
A chave pública então é composta pelos pares $(a_i, b_i = \frac{\langle a_i, s \rangle}{q} + e_i)$

ENCRIPTAÇÃO

Escolher um bit plaintext $b \in \{0, 1\}$;
Tome um subconjunto S dos pares (a_i, b_i);
O bit codificado é o par $(\sum_{i \in S} a_i, x/2 + \sum_{i \in s} b_i)$

DECRIPTAÇÃO

Tendo (a, b), obtemos $b =$
$round(b - \frac{\langle a_i, s \rangle}{q})$, arredondando para 0 ou para 1 conforme apropriado.

5 CRIPTOANÁLISE

Normalmente, o objetivo em atacar um sistema criptográfico é recuperar a chave em uso, em vez de simplesmente recuperar o texto claro de um único texto cifrado. Os ataques criptoanalíticos contam com a natureza do algoritmo e talvez informações acerca do texto original, ou até mesmo do texto cifrado. Explorando esses pontos, o método tenta - no caso dos criptossistemas de reticulados, por meio dos processos de redução que serão apresentados nesta seção - deduzir a chave utilizada.

Os criptossistemas baseados em reticulados são vistos como seguros, em primeira análise, devido à dificuldade em resolver os problemas SVP e CVP pelo uso de uma base ruim utilizando o algoritmo de Babai. Uma possível solução para isto seria criar um algoritmo que, dada uma base ruim, tivesse como saída uma base boa que descrevesse o mesmo reticulado. Este processo de fato existe, e denomina-se redução de reticulado. Serão apresentados um algoritmo base, para o caso mais simples (bidimensional) e três generalizações do mesmo para maiores dimensões, cada um com a intenção de apresentar uma base melhor (com possível aumento de custos computacionais devido à maior acurácia demandada).

5.1 REDUÇÃO GAUSSIANA

Para reticulados de duas dimensões, a redução gaussiana é capaz de resolver o problema do SVP. A idéia é basicamente subtrair múltiplos de um dos vetores da base, até que a operação não seja mais possível [5].

Seja, então, um reticulado $\mathcal{L} \subset \mathbb{R}^2$ com bases v_1 e v_2, tais que $||v_1|| < ||v_2||$. Agora vamos calcular o vetor v_2^*, projeção de v_2 sobre a direção ortogonal a v_1.

$$v_2^* = v_2 - \frac{v_1.v_2}{\|v_1\|^2} v_1 \tag{5.1}$$

No entanto, o vetor v_2^* provavelmente não pertence a $\mathcal{L}$. Então, fazemos o vetor v_2^* como:

$$v_2^* = v_2 - mv_1, \qquad m = \left\lfloor \frac{v_1.v_2}{\|v_1\|^2} \right\rceil v_1 \tag{5.2}$$

Se $m = 0$, v_2^* é o menor vetor não nulo de $\mathcal{L}$.

Caso contrário, substituimos v_1 por v_2^*e repete-se o processo, até que m seja igual a 0. A base resultante do algorítmo será tão próxima da ortogonalidade quanto possível para os parâmetros do reticulado.

5.2 REDUÇÃO LLL

A redução Gaussiana é eficaz na redução de reticulados bidimensionais, mas não é trivialmente generalizada para dimensões maiores.

O algoritmo LLL (homenagem aos seus criadores: Lenstra, Lenstra e Lovász) em combinação com o algortimo de Babai busca resolver o SVP com um fator de aproximação de $2^{\frac{n-1}{2}}$ em tempo polinomial [14], além de retornar uma base boa. Isso permite, em muitos casos, a resolução imediata de problemas do tipo SVP e CVP em reticulados de dimensões pequenas.

O funcionamento do LLL consiste em minimizar os comprimentos das projeções de cada elemento da base sob o subespaço gerado pelas anteriores por meio de uma troca semelhante à vista na seção anterior.

Além disso, uma nova condição é imposta para que o algoritmo funcione em tempo polinomial, chamada condição de Lovász [19].

$\frac{||b_i||^2}{||b_{i-1}||^2} \geq \delta - \mu_{i,i-1}^2$, para $\delta \in (1/4, 1)$ para cada i.

O algoritmo LLL podem ser resumido como uma aplicação do algoritmo de Gauss iterativamente a cada par de vectores (b_i, b_{i-1}), assegurando que cada par de vetoress satisfaça a condição Lovász. O algoritmo também assegura que a base é de tamanho reduzido em todos os momentos. Quando trocamos os dois vetores (b_i, b_{i-1}), pode-se arruinar a condição de Lovász, portanto cada vez em que se trocam dois vetores, diminuímos i por um para ver se é preciso corrigir alguma coisa para os valores anteriores do i. Note-se que tomar $d = 2$ e $\delta = 1$ leva ao algoritmo de Gauss bidimensional, de modo que o Algoritmo LLL poderia ser visto como uma generalização do algoritmo de Gauss. A descrição do algoritmo LLL é simplificada de qualquer implementação real do algoritmo, onde deixamos os detalhes sobre a

atualização dos vetores Gram-Schmidt e coeficientes. Pode-se assumir que a ortogonalização de Gram-Schmidt é sempre conhecida e é atualizada em relação à base atual.

Pode-se provar que o algoritmo LLL é executado em tempo polinomial, e atinge aproximação exponencial certa a menos de fatores de Hermite [20].

Algoritmo 5 (Algoritmo de Redução de Base (LLL))**.**

```
Requer uma base {b1, ..., bd} ∈ L e uma constante δ ∈ (1/4, 1)
i ← 2
while i ≤ d do
    b_i ← b_i − Σ_{j=1}^{i−1} ⌊μ_{i,j}⌉ b_j
    if ||b_i^*||^2 ≥ (δ − μ_{i,j}^2) ||b_{i−1}^*||^2 then
        i ← i + 1
    else
        swap (b_i, b_{i−1})
        i ← max {2, i − 1}
    end if
end while
```

5.3 KZ

Nas duas seções anteriores, foram apresentados algoritmos de redução de bases bidimensionais: algoritmo de Gauss, para encontrar a melhor base bidimensional e o algoritmo LLL, para encontrar bases em dimensões elevadas que são localmente semelhantes a uma redução Gaussiana. Agora, será apresentada uma generalização ainda maior do algoritmo de redução com blocos k-dimensionais, com $k \geq 2$. Primeiro, deve ser caracterizado o que significa uma base ser ideal, e então fornecer um algoritmo para encontrar tais bases em dimensão k. Neste caso, utiliza-se uma subrotina de redução de blocos como o LLL para obter bases de reticulados em dimensões altas com menores fatores de aproximação exponencial que o algoritmo LLL numa quantidade razoável de tempo.

Para encontrar [2] bases ideais em dimensão k, primeiro é necessário decidir que tipo de otimização é desejada. Para conseguir uma boa noção de redução, por ora simplesmente será assumido que há um oráculo $\mathcal{O}$ que resolve o problema do menor vetor para qualquer k. Algoritmos reais que podem atuar como oráculo para o SVP existem, mas não serão

discutidos neste trabalho. Uma escolha imediata de otimalidade seria de exigir que uma base k-dimensional ideal reduzida deve satisfazer $\|b_i\| = SVP_i(\mathcal{L})$, para cada i de 1 a k. Mas atingir isso parece difícil, mesmo com acesso a oráculos SVP. Com oráculos SVP, só é possível encontrar um vetor mais curto em qualquer estrutura, e assim a noção de redução em função do primeiro mínimo faria mais sentido. Abaixo está descrita uma noção de redução, juntamente com um algoritmo para alcançá-la, que dará bases com aproximação muito pequena. Primeiro, com acesso a um oráculo SVP, podemos facilmente deixar o menor vetor base b1 como a solução para o SVP $\|b_1\| = SVP_1(\mathcal{L})$, escolhendo o vetor base primeira saída como uma menor vetor de L. Então, os vetores $v \in \mathcal{L}$ são decompostos da seguinte forma: $v = v_1 + v_2$, $v_2 = \alpha_1.b_1$ com uma combinação linear de b_1, e $v_2 \in \langle b_1 \rangle^{\perp}$ ortogonal a b_1. O conjunto destes vetores v_2 é também representado por $\pi_2(\mathcal{L})$, o reticulado projetado de $\mathcal{L}$ sobre a complemento da expansão de $\langle b_1 \rangle$. Vetores em $\pi_2(\mathcal{L})$ não pretencem, em geral, ao reticulado, mas pode-se sempre levantar qualquer vetor em $\pi_2(\mathcal{L})$ para um vetor do reticulado, somando uma quantidade adequada de b_1 a ele. Com a técnica de redução de tamanho, esta quantidade adequada sempre pode ser escolhida entre $-\frac{1}{2}$ e $+\frac{1}{2}$. Um vetor pequeno em $\pi_2(\mathcal{L})$ não corresponde necessariamente a um vetor pequeno do reticulado, mas para bases de tamanho reduzido, a seguinte desigualdade é verdadeira:

$$\|b_i\|^2 = \left\| b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^* \right\|^2 \leq \|b_i^*\|^2 + \sum_{j=1}^{i-1} |\mu_{i,j}| \, \|b_i^*\|^2 \leq \|b_i^*\|^2 + \frac{1}{4} \sum_{j=1}^{i-1} \|b_i^*\|^2 \tag{5.3}$$

Em vez de encontrar um vetor que alcança o segundo mínimo sucessivo de $\mathcal{L}$, utiliza-se o oráculo SVP no reticulado $\pi_2(\mathcal{L})$ para encontrar um vetor mais curto neste reticulado projetado, tratando esse vetor mais curto como uma projeção b_2^* de algum vetor b_2 no reticulado. Já que a elevação garante que a base $\{b_1, b_2\}$ é de tamanho reduzido e $b_1 = b_1^*$ é um vetor mais curto de $\mathcal{L}$, pode-se utilizar 5.3 para obter um limite superior para o comprimento do b_2 elevado como se segue:

$$\|b_2\|^2 \leq \|b_2^*\|^2 + \frac{1}{4} \|b_1^*\|^2 = SVP_1^2(\pi_2(\mathcal{L})) + \frac{1}{4} SVP_1^2(\pi_2(\mathcal{L})) \tag{5.4}$$

Uma vez que o vetor mais curto no reticulado projetado $\pi_2(\mathcal{L})$ nunca é maior do que o vetor obtido do segundo mínimo de $SVP_2(\mathcal{L})$, tem-se: $SVP_1(\mathcal{L}),\ SVP_1(\pi_2(\mathcal{L})) \leq SVP_2(\mathcal{L})$. Então, dividindo ambos os lados por $SVP_2^2(\mathcal{L})$ e usar esses limites inferiores em $SVP_2(\mathcal{L})$, obtém-se.

$$\frac{\|b_2\|^2}{SVP_2^2(\mathcal{L})} \leq \frac{4}{3} \tag{5.5}$$

Depois de encontrar b_2 com $\|b_2\| = SVP_1(\pi_2(\mathcal{L}))$, repete-se o processo em $\pi_3(\mathcal{L})$, que consiste dos vetores projetados do reticulado em $\langle b_1, b_2\rangle^{\perp}$. Após aplicar o oráculo e realizar a redução, obtemos:

$$\frac{\|b_i\|^2}{SVP_i^2(\mathcal{L})} \leq \frac{i+3}{3} \tag{5.6}$$

Esta noção de redução, em que os vetores Gram-Schmidt de uma base $\{b_1, ..., b_k\}$ satisfazem $\|b_i^*\| = SVP_1(\pi_i(\mathcal{L}))$ para cada i, é também chamado de redução de Korkine-Zolotarev (KZ) [1] e as bases são chamados KZ-reduzidas. O procedimento acima descrito, para se obter bases KZ-reduzidas, está resumido no algoritmo abaixo. O teorema seguinte resume a qualidade do vetor base em primeiro lugar, e a razão entre os comprimentos do primeiro e do último vetores de Gram-Schmidt de qualquer base KZ-reduzida.

Teorema 5.1. *Dada uma base $\{b_1, ..., b_k\}$ de um reticulado L e oráculo de SVP $\mathcal{O}$ para até k dimensões, o algoritmo de redução Korkine-Zolotarev termina após a maioria das chamadas k para $\mathcal{O}$ e gera uma base $\{b_1, ..., b_k\}$ reduzida de L satisfazendo:*
$\frac{\|b_1\|}{SVP_1(L)} = 1$, $\frac{\|b_1\|}{vol(L)} \leq \sqrt{H_k} = O\left(\sqrt{k}\right)$, $\frac{\|b_1^*\|}{\|b_k^8\|} \leq k^{(1+lnk)/2}$
onde H_k é o conjunto de todas as bases k-dimensionais KZ-reduzidas..

Algoritmo 6 (Algoritmo de redução KZ de base)**.**

Requer: uma base $\{b_1, ..., b_k\}$ de L, e um $\mathcal{O}$ SVP-oráculo para até dimensões k
for $i = 1$ a k

chamar o $\mathcal{O}$ para encontrar um vetor $b_i^* \in$
$\pi_i(\mathcal{L})$ de comprimento $SVP_1(\pi_i(\mathcal{L}))$

elevar b_i^* dentro de um vetor b_i do reticulado tal que $\{b_1, ..., b_k\}$
seja de tamanho reduzido.

substituir os vetores da base $\{b_{i+1}, ..., b_k\}$ por vetores do reticulado
$\{b_{i+1}, ..., b_k\}$ tal que $\{b_1, ..., b_k\}$ é um base para $\mathcal{L}$

end for

Note que encontrar bases KZ-reduzidas é pelo menos tão forte quanto encontrar o menor vetor em k dimensões, desde que o vetor mais curto for o menor vetor da reticulado. Assim,

em dimensões elevadas este algoritmo e sua redução são impraticáveis. Este algoritmo só termina num período de tempo razoável quando k é suficientemente pequeno. Para encontrar bases ideais para arbitrários d-dimensionais reticulados, para d alta, são necessários métodos diferentes.

5.4 BKZ

O algoritmo KZ apresentado na seção anterior pode ser usado como uma subrotina para encontrar bases d-dimensionais boas. Se cada bloco de k bases consecutivas for KZ-reduzida, então pode-se provar [28] que a base b_1 satisfaz a seguinte condição:

$$\frac{\|b_1\|}{\lambda_1(L)} \leq \alpha_k^{(d-1)/(2k-2)} \tag{5.7}$$

A prova é feita comparando os comprimentos dos pares de vetores $b_{i(k-1)}$ e $b_{(i+1)(k-1)-1}$. Para cada par, utiliza-se o fato de que o bloco contendo esses vetores como primeiro e último vetores é KZ-reduzido.

Usando o fato de cada bloco ser KZ-reduzido, Schorr e Euchner propuseram um algoritmo conhecido como Block Korkine-Zolotarev (BKZ) similar ao LLL, em que itera-se a redução KZ em blocos locais até cada bloco ser completamente reduzido. Deve-se voltar para as $k-1$ posições anteriores e, caso as bases não estejam mais KZ-reduzidas, o algoritmo é executado de novo até que todas as k bases estejam KZ-reduzidas.

Algoritmo 7 (Algoritmo de redução BKZ)**.** *Requer: uma base* $\{b_1, ..., b_d\}$ *de L, blocos de tamanho k, uma constante* $\delta \in (1/4, 1)$ *e um* $\mathcal{O}$ *SVP-oráculo para até dimensões k.*

```
while (ocorrer mudança nenhuma)
    for i = 1 a d − k + 1
        reduzir a base Π_i (b_i, ..., b_{i+k−1}) pelo algoritmo KZ
        reduzir o tamanho da base {b_1, ..., b_d}
    end for
end while
```

Para o LLL, pode-se provar que ele roda em tempo polinomial. Mas para o BKZ, a prova para a análise de complexidade não pode ser feita. No entanto, simulações sugerem que o algoritmo roda em tempo polinomial, à exceção da execução do KZ em si, que ainda não é bem compreendido [21]. Apesar disso, pode-se mostrar que o vetor da primeira base satisfaz condições de tamanho.

Teorema 5.2. *Dada uma base $\{b_1, ..., b_d\}$ de um reticulado L, e um $\mathcal{O}$ SVP-oráculo para até dimensões k, o algorimo de redução BKZ retorna uma base $\{b_1, ..., b_d\}$ que satisfaz:*

$$\frac{\|b_1\|}{\lambda_1(L)} \leq \left(k^{\frac{1+lnk}{2k-2}}\right)^{d-1} \tag{5.8}$$

5.5 BKZ 2.0

Algumas melhorias foram propostas para o BKZ, a maioria envolve melhorar a subrotina do SVP. Os melhores resultados foram obtidos por Chen e Nguyen [24] e originaram o BKZ 2.0. Técnicas de enumeração melhoradas foram usadas para aumentar a velocidade do algoritmo e fazer com que o BKZ funcionasse com blocos muito grandes. Devido aos grandes blocos, a enumeração foi feita em projeções de reticulados de grandes dimensões. Essas projeções se comportam como reticulados aleatórios [24]. Assim, LLL é utilizado para remover as dependências lineares criadas e a enumeração será aplicada em uma base já reduzida pelo LLL.

A ideia de enumeração envolve tomar todas as combinações possíveis de vetores das bases do reticulado e encontrar qual deles é menor. Como existem infinitas combinações, é necessário estabelecer uma enumeração de vetores das bases menores do que um certo valor. Um modo de ver esta enumaeração é como uma busca em uma árvore, na qual cada nó corresponde a um vetor, com a condição que a norma de cada vetor seja menor do que um parâmetro R (de modo a limitar as possibilidades e não criar bases arbitrariamente ruins). O i-ésimo nível da árvore corresponde aos vetores de $\pi_{d-i+1}(L)$, para $0 \leq i \leq d$, e os filhos de cada vetor v na árvore são os vetores $u \in \pi_{d-i+2}(L)$ que satisfazem $v = \pi_{d-i+1}(u)$

Outra melhoria é terminar o BKZ antes que o algoritmo termine. Na prática, estágios intermediários do algoritmo levam a bases que estão perto das BKZ-reduzidas [29].

Além disso, fora observado por Chen e Nguyen que as melhorias no BKZ melhoram, de fato, a base retornada pelo algoritmo e a aleatoriedade observada nas projeções permitiu a

Chen e Nguyen analizarem o BKZ não somente experimentalmente, como também teoricamente.

As ideias básicas para a simulação são usar um bloco de tamanho B, um número de iterações N e os comprimentos dos vetores da base do algoritmo de Gram-Schimdt $log\left(||b_i^*||\right)$.

6 AVALIAÇÃO DE SEGURANÇA

Tendo sido descritos tanto os criptossistemas em reticulados quanto as técnicas de redução nos mesmos, faz-se necessário procurar o quão efetivas são as técnicas de redução quando aplicadas a um dos sistemas.

Algumas questões são levantadas para responder quanto esforço é necessário para se quebrar um criptossistema baseado em reticulado. Qual dos métodos, LLL ou BKZ, deve ser usado para um ataque mais eficiente? O comportamento dos algoritmos de redução nem sempre é bem entendido. O LLL parece ser capaz de resolver boa parte das instâncias do SVP, apesar dele ser um problema difícil. E por último, ainda não se sabe muito a respeito do tempo de funcionamento do BKZ. Aparentemente, ele parece ser o melhor algoritmo de redução de bases [2], resultando em bases mais precisas, pagando por isso com um maior custo de complexidade no tempo.

Observando o trabalho de Gama e Nguyen [21], percebe-se que o mesmo considera o desempenho prático de algoritmos de redução de base. No entanto, o seu trabalho não foi especificamente destinado a criptografia, por isso deve-se considerar primeiramente o trabalho por Ruckert e Schneider [22], que adaptou a abordagem por Gama e Nguyen para a análise de quebrar rede baseados em sistemas criptográficos. A fim de criar um quadro abrangente com base em ambos o LWE e os problemas do SIS, Ruckert e Schneider assumem que a melhor maneira de resolver LWE é, reduzindo o problema a SIS e aplicando um algoritmo de redução. Alguns trabalhos de Lindner e Peikert sugerem, contudo, que este poderia não ser a estratégia ideal para um atacante [23].

6.1 GERAÇÃO DE RETICULADOS

Para determinar a performace dos algoritmos de redução na prática, é necessária a geração de reticulados para os experimentos. Ainda não se sabe que tipos de bases devem ser usadas para avaliar com precisão a performace dos algoritmos de redução.

Uma vez gerados aleatoriamente os reticulados, agora é necessário representá-los como um conjunto de vetores (base). Como algumas dessas bases podem influenciar a performace dos algoritmos de redução, Gama e Nguyen escolheram bases aleatórias para evitar tais influen-

cias. Como não existe um padrão para caracterizar as bases de um reticulado, consideram-se aleatórias bases com vetores muito grandes, escolhidos por uma heurística. Apesar do procedimento de geração de bases não ser explicitado em seu trabalho [21], há um artigo [7] que detalha a heurística para escolher essas bases aleatórias.

Inspirados pelos resultados de Gama e Nguyen, Rücket e Schneider analisaram a segurança dos criptossistemas baseados em reticulados desejando criar uma estrutura para determinar a segurança dos criptossistemas SIS e LWE na prática.

6.2 COMO MEDIR A SEGURANÇA

Essa estrutura de avaliação do nível de segurança segue três passos:

- Representar a dificuldade dos problemas SIS e LWE com um único parâmetro.
- Realizar experimentos relacionando esse parâmetro ao esforço para realizar o ataque.
- Aplicar essa estrutura a diversos esquemas criptográficos para medir e comparar seus níveis de segurança.

Para medir a segurança, precisa-se levar em consideração a capacidade que possui um possível atacante e as possíveis evoluções tanto em potência computacional quanto em métodos criptoanalíticos. Esse valor da capacidade do atacante será medido em dólar-dias, em tradução livre do inglês *dollar-days* [30], que representa o custo do equipamento, em dólares, multiplicado pelo tempo que o atacante leva para concluir um ataque, em dias.

Considere um sistema criptográfico com parâmetro de segurança k. Suponha que o melhor ataque conhecido contra esse criptossistema requer $t(k)$ segundos em um computador de d dólares. O custo deste ataque representado em dólar-dias é dado por:

$$T(k) = \frac{d * t(k)}{3600 * 24} \tag{6.1}$$

Se uma estimativa da função $T(k)$ é conhecida, podem-se escolher parâmetros k da seguinte forma: considere que um atacante disponha de T_{y_0} dólares, em que y_0 representa algum ano.. Para garantir segurança contra esse atacante, o parâmetro k precisa exceder um valor k^* tal que $T(k^*) \geq T_{y_0}$.

Também é possivel considerar futuras tecnologias e estimar qual será a segurança do sistema. Para isso, deve-se considerar a regra chamada lei dupla de Moore [30], do inglês *double Moore law*. Esta lei estabelece que o poder computacional dobra a cada 18 meses. Além disso, ela diz que a segurança de um criptossistema decai por um fator de $2^{-12/9}$. Contudo, esta função é baseada no avanço dos algoritmos na área de fatoração de números inteiros. Este fator fora adotado também para a criptoanálise dos reticulados por não haver ainda nenhum estudo sobre o valor desse fator. Logo, para ter segurança até um ano y contra um atacante que tem T_{y_0} dólares-dia no ano y_0, o parâmetro de segurança deve satisfazer:

$$T(k) \geq T_{y_0} * 2^{(y-y_0)*12/9} \tag{6.2}$$

6.3 SEGURANÇA DO LWE E SIS

Os criptossistemas SIS e LWE possuem diversos parâmetros que influenciam nas suas seguranças. Ainda assim, é desejado representar a segurança desses criptossistemas com um único parâmetro, que corresponde ao melhor ataque conhecido [22].

Da descrição do SIS no capítulo 3, tem-se que o SIS possui quatro parâmetros para o reticulado:

$$\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m |\ Ax = 0 \mod q\} \tag{6.3}$$

Em que dois deles representam a dimensão da matriz utilizada, um é o inteiro q que serve de módulo e o último é o vetor que representa a cota superior. Para resolver o sistema indeterminado $Ax = 0$, fixam-se k coordenadas de x como sendo iguais a 0, e o sistema é resolvido nas outras coordenadas. Isso leva a um subreticulado menor, mas que continua possuindo aproximadamente o mesmo volume. Um algoritmo de resolução de reticulado é então aplicado no subreticulado $\Lambda_q^{\perp}(A')$, onde A' é obtida de A pela remoção completa de k de suas colunas. Pode-se demonstrar que o algoritmo de redução deve ser capaz de resolver o $SVP\gamma$ para um $\gamma \leq \sqrt[d]{v/q^{n/d}}$, onde d é o número de colunas de A'[2].

Análises experimentais do BKZ determinam que o parâmetro mais influente na performance do algoritmo na resolução de um reticulado SIS é o fator de aproximação requerido γ. Ajustes de γ de modo que ele seja menor do que o máximo que poderia ser adquirido por um possível atacante tornariam o sistema seguro por uma certa quantidade de tempo.

Uma cota superior para a segurança do LWE pode ser obtida transformando-se o reticulado do SIS no reticulado modular LWE, o que mostra que o SIS é ao menos tão seguro

quanto o LWE. Contudo, não foram encontrados estudos diretamente sobre a segurança do LWE que não utilizassem esta equivalência com o SIS para encontrar um ataque.

7 CONCLUSÃO

Dos estudos apresentados acerca dos diferentes criptossistemas baseados em reticulados, pode-se, de imediato, concluir que alguns deles são de implementação inviável, devido tanto a problemas de tamanho de chave muito grande em comparação com outros criptossistemas atualmente em uso (como é o caso do GGH, que tem tamanho de chave inviável nas dimensões nas quais ele se torna seguro), quanto a vulnerabilidades ao ataque pelos algoritmos LLL e KZ, o que torna o sistema inerentemente inseguro. Deste modo, também pode-se perceber que nem todos os problemas inicialmente tidos como difíceis o são, uma vez que não necessariamente é preciso possuir a informação de arapuca para resolvê-los em tempo hábil por um ataque de redução.

Conclui-se também que deve haver cuidado ao se selecionar um problema mesmo que ele aparente ser difícil. Deste modo, concluímos que o GGH em sua forma atual não seria viável como criptossistema, e que, salvo por meio de uma escolha cuidadosa de parâmetros, o NTRU também se veria vulnerável. O NTRU, devido a sua baixa complexidade algoritmica de encriptação e decriptação, poderia ainda ser utilizado em aplicações que priorizassem a velocidade acima da segurança [31].

Em contrapartida, alguns dos problemas apresentados mostraram-se resistentes à redução LLL básica, caso do SIS e do LWE. Isso indica que ainda há grande possibilidade de estudos na área de redução de reticulados de modo a resolver estes problemas. Estudos neste sentido já foram iniciados[24] [22] de modo a produzir métodos mais eficientes de redução, o que poderia comprometer a aparente segurança de criptossistemas baseados no LWE ou no SIS. Apesar disso, o LWE e o SIS mostram-se promissores em resistência a ataques de redução com os métodos apresentados, mesmo considerando o aumento do poder de processamento com o tempo. Modificações significativas de performance no BKZ 2.0 podem, contudo, servir para diminuir esta segurança.

Um dos problemas encontrados na avaliação da segurança é o fato de a teoria de redução de reticulados não estar madura o suficiente de modo a poder prever tanto seu tempo de execução quanto sua precisão ao encontrar melhores bases, o que faz com que seja apenas possível realizar a avaliação de segurança por meio da implementação direta dos algoritmos. No caso do LWE, por ser um algoritmo probabilístico de aprendizado de máquina complexo, há grande dificuldade de implementação do mesmo, o que impede em muitos casos o teste da segurança do criptossistema ante um ataque. No caso do SIS, as implicações de um possível

ataque de redução são melhor compreendidas, e, no estado atual da teoria de redução, o SIS encontra-se relativamente seguro em relação a este tipo de ataque. Outros possíveis métodos de ataque não se mostraram muito promissores, sendo deixados de lado.

Para trabalhos futuros, reforça-se a necessidade da implementação e teste dos algoritmos, tanto em suas versões básicas como com as modificações apresentadas de modo a de fato realizar uma avaliação contundente sobre a segurança dos criptossistemas, tanto atualmente como utilizando previsões para o poder computacional de gerações futuras.

8 REFERÊNCIAS

[1] Ruckert, Markus; Schneider, Michael. Estimating the Security of Lattice-based Cryptosystems. Technische Universitat Darmstadt, Germany, 2010.

[2] Laarhovem, Thijs; van de Pol, Joop, Wegner, Benne de. Solving Hard Lattice Problems and the Security of Lattice-Based Cryptosystems, 2012.

[3] Shor, Peter W. (1997), "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SIAM J. Comput. 26 (5): 1484–1509

[4] CARAGEORGE, Edward C., Criptografia Baseada em Reticulado. Instituto Militar de Engenharia, Rio de Janeiro, 2012.

[5] HOFFSTEIN, Jeffrey; PIPHER, Jill; SILVERMAN, Joseph H. An Introduction to Mathematical Cryptography, 2008, Springer.

[6] O Goldreich, S Goldwasser, and S Halevi. Public-key cryptosystems from lattice reduction problems. CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, Jan 1996.

[7] O. Goldreich, S. Goldwasser and S. Halevi, Collision-Free Hashing from Lattice Problems, Electronic Colloquium on Computational Complexity, vol. 3, no. 42, pp. 1–10, 1996.

[8] AJTAI, M.; DWORK, C. A public-key cryptosystem with worstcase/ averagecase equivalence. In: Proc. 29th ACM STOC. [S.l.: s.n.], 1998. p. 284–293

[9] Miklós Ajtai. Generating Hard Instances of Lattice Problems. In STOC, pages 99-108. ACM, 1996.

[10] Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In Mitzenmacher, pages 169-178.

[11] Daniele Micciancio. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient one-way Functions. Computational Complexity, 16(4):365-411, 2007. Prelim. in FOCS 2002.

[12] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring-based public key cryptosystem. In Algorithmic Number Theory (Portland, OR, 1998), volume 1423 of Lecture Notes in Comput. Sci., pages 267–288. Springer, Berlin, 1998.

[13] REGEV, O. On lattices, learning with errors, random linear codes, and cryptography. In: Proc. 37th ACM Symposium on Theory of Computing (STOC). [S.l.: s.n.], 2005. p. 84–93.

[14] A. K. Lenstra, H. W. Lenstra, Jr., and L. Losvász. Factoring polynomials with rational coefficients. Math. Ann., 261(4):515–534, 1982.

[15] STALLINGS, William. Criptografia e Segurança de Redes - Princípios e Prática, 2007, PEARSON, Prentice Hall, São Paulo

[16] D. Micciancio and O. Regev, Worst-Case to Average-Case Reductions Based on Gaussian Measures, SIAM J. Comput., vol. 37, no. 1, pp. 267–302, 2007.

[17] O. Regev, On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, Proc. 37th Symp. Theory of Computing (STOC), pp. 84–93, 2005.

[18] C.F. Gauss, Disquisitiones Arithmeticae, Leipzig, 1801.

[19] L. Lovász. An algorithm theory of numbers, graphs and convexity, CBMS-NSF Reg. Conf. Ser. Appl. Math, vol 50, pp. 91, 1986

[20] C. Hermite, Extraits de lettres de M. Hermite à M. Jacobi sur diff'erent objets de la théorie des nombres, deuxième lettre, J. Reine Angew. Math. 40 [1850], 279–290.

[21] N. Gama and P.Q. Nguyen, Predicting Lattice Reduction, Proc. 27th Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 31–51, cn2008.

[22] M. Rycket and M. Schneider, Estimating the Security of Lattice-based Cryptosystems, Cryptology ePrint Archive, Report 2010/137, pp. 1–33, 2010.

[23] R. Lindner and C. Peikert, Better Key Sizes (and Attacks) for LWE-based Encryption, Topics in Cryptology (CT-RSA), pp. 319–339, 2011.

[24] Y. Chen and P.Q. Nguyen, BKZ 2.0: Better Lattice Security Estimates, Proc. 17th Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 1–20, 2011.

[25] A. Korkine and G. Zolotarev, Sur les formes quadratiques, Math. Ann., vol. 6, no. 3, pp. 366–389, 1873.

[26] E. Rieffel and W. Polak, An Introduction to Quantum Computing for Non-Physicists, ACM Computing Surveys, Vol. 32, No. 3, September 2000C.P.

[27] Schnorr and M. Euchner, Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems, Mathematical Programming, vol. 66, no. 2–3, pp. 181–199, 1994.

[28] C.P. Schnorr, A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms, Theoretical Computer Science, vol. 53, no. 2–3, pp. 201–224, 1987.

[29] G. Hanrot, X. Pujol and D. Stehl´e, Analyzing Blockwise Lattice Algorithms using Dynamical Systems, Proc. 31th Cryptology Conference (CRYPTO), pp. 447–464, 2011.

[30] A.K. Lenstra, Key Lengths, in The Handbook of Information Security, ch. 114, 2005.

[31] A. C. Atici, L. Batina, J. Fan and I. Verbauwhede. Low-cost implementations of NTRU for pervasive security, 19th IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp79-84, 2008.