ON LATTICE BASED DIGITAL SIGNATURE SCHEMES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FARID JAVANI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

JULY 2014

Approval of the thesis:

ON LATTICE BASED DIGITAL SIGNATURE SCHEMES

submitted by **FARID JAVANI** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** ______________

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** ______________

Prof. Dr. Ersan Akyidiz
Supervisor, **Cryptography, METU** ______________

Examining Committee Members:

Prof. Dr. Ersan Akyildiz
Cryptography, METU ______________

Prof. Dr. Ferruh Özbudak
Cryptography, METU ______________

Dr. Muhiddin Uğuz
Cryptography, METU ______________

Dr. Murat Cenk
Cryptography, METU ______________

Dr. Hamdi Murat Yıldırım
Computer Tech. and Information Sys., Bilkent University ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: FARID JAVANI

Signature :

ABSTRACT

ON LATTICE BASED DIGITAL SIGNATURE SCHEMES

Javani, Farid
M.S., Department of Cryptography
Supervisor : Prof. Dr. Ersan Akyidiz

July 2014, 46 pages

Lattice based cryptography is one of the few hopes for secure public key cryptography in post quantum era since there is no known polynomial time quantum algorithm that can solve hard lattice problems. But despite this precious property, for a cryptographic construction which is designed based on a hard lattice problem, to be secure, required time and space is not efficient. This has led to introduction of structured lattices that need less time and space; indeed the only existing standard on lattice based cryptography is based on hardness of solving lattice problems for a class of structured lattices, called NTRU lattices; and though it lacks a security proof, in terms of efficiency this standardized cryptographic system can be compared to cryptographic constructions which are based on Integer Factorization Problem or Discrete Logarithm Problem.

Digital signatures are important cryptographic primitives that can naturally be designed using hard lattice problems. In this thesis we have studied three signature schemes that are based on hardness of solving certain lattice problems; first scheme is an efficient signature scheme with provable security, the second scheme is GGH signature and the third one is NTRUSign. We also have studied a brilliant cryptanalysis technic which is applicable on GGH signature and NTRUSign and implemented it on a lattice of dimension 15.

Keywords : Public key Cryptography, Lattice based Cryptography, Digital Signatures, Basis Reduction Algorithms

ÖZ

KAFES TABANLI DIJITAL İMZALAR ÜZERINE

Javani, Farid
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ersan Akyildiz

Temmuz 2014, 46 sayfa

Kafes tabanlı şifreleme, kuantum sonrası çağda güvenli açık anahtar algoritmalarını oluşturmak için gerekli olan az sayıdaki araçlardan birisidir. Bunun nedeni ise zor kafes problemlerini çözebilecek polinomiyal zamanlı kuantum algoritmalarının henüz mevcut olmamasıdır. Zor kafes problemlerine dayanan kriptografik yapılar, bu önemli özelliğine rağmen zaman ve kapladığı alan açısından çok verimli değildir. Bu durum daha az zaman ve yer gerektiren yapılı kafeslerin oluşturulmasına yol açmıştır. Kafes tabanlı şifreleme sistemlerindeki var olan tek standart, bu yapılı kafesler üzerinde tanımlanan zor problemlere dayanmaktadır. NTRU olarak adlandırılan bu standart, verimlilik açısından sayılar teorisine dayanan şifreleme sistemleriyle kıyaslanabilir.

Kriptografik açıdan büyük öneme sahip olan dijital imzalama algoritmaları kafes tabanlı zor problemler kullanılarak da oluşturulabilir. Bu tezde, kafes tabanlı zor problemlere dayanan üç farklı dijital imzalama algoritması çalışılmıştır. Bu algoritmalar sırasıyla ispatlanabilir güvenliği olan bir imzalama algoritması, GGH imzalama algoritması ve NTRUSign imzalama algoritmasıdır. Bunlara ek olarak, GGH ve NTRUSign algoritmalarının kriptoanalizi de incelenmiştir.

Anahtar Kelimeler : Açık Anahtarlı Şifreleme, Kafes Tabanli Şifreleme, Dijital İmza, Kafes İndirgeme Algoritmaları

To my mother

ACKNOWLEDGMENTS

I am very grateful to my supervisor, mentor and teacher Professor Dr. Ersan Akyildiz, for writing this thesis was possible only because of his great knowledge, support and motivation. I would also thank him for all his support and consideration in every step of my studies at METU during the past three years. I will always consider meeting and knowing him as a precious and lucky incident of my academic life.

TABLE OF CONTENTS

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Integer Factorization Problem (IFP) and Discrete Logarithm Problem (DLP) are considered to be strong trapdoors for cryptographic constructions when dealing with algorithms and attacks that are applicable by classical computers. There is no known polynomial time algorithm to solve IFP or DLP thought also there is no mathematical proof on hardness of either of these problems. Nonexistence of efficient algorithms to solve IFP and DLP has lead to systems which are built based on their hardness. RSA, for example, is a widely used asymmetric cryptosystem based on IFP and Digital Signature Algorithm is based on DLP. But none of these systems are secure when algorithms and attacks applicable by quantum computers are available. Shor in [24] proposes algorithms that can successfully attack cryptosystems that are based on IFP and DSP that only require polynomially many steps. Moreover, based on Shor's algorithm for discrete log, an algorithm to solve Elliptic Curve Discrete Logarithm Problem on GF(p) have also been proposed in [20]. But compared to classic approaches towards solving hard lattice problems, Shortest Vector Problem (SVP) for example, superiority of quantum algorithms is not considerable. In [11], Ludwig proposes a quantum lattice reduction algorithm that approximates SVP within a factor of $(k/6)^{n/2k}$ with the estimated running time almost square root of previously know best reduction algorithm. It is also worth mentioning that no such threats exist for classical symmetric cryptography; that is, attacks on symmetric key cryptographic systems using quantum computers does not have any advantages over attacks that are performed by classical computers [3].

Though yet there does not exist any practical quantum attacks to be considered as applicable threats on number theoretic cryptographic systems, it has become obvious that for cryptography to survive in quantum computation era, it is necessary to discover and use constructions and cryptographic primitives that do not tend to have weaknesses against quantum algorithms. This serious concern has caused a new area to emerge which is called *post-quantum cryptography*. Hash-based cryptography, Code-based cryptography, Lattice-based cryptography and Multivariate-quadratic-equations cryptography are four families of cryptographic systems that are believed to resist quantum cryptanalysis (for details see [3]). In this thesis we will study Lattice based cryptography, to be more specific, different design approaches of signatures based on lattice problems.

1.1 Lattice-Based Cryptography

Origin of lattice-based cryptography is Ajtai's discovery [1] of the relation between average case and worst case hardness of lattice problems. In his seminal work, Ajtai showed that if one can find a solution $\mathbf{x}$ to the $\mathbf{Mx} \equiv 0 \bmod q$ with length less than n for a uniformly random chosen matrix $\mathbf{M} \in \mathbb{Z}^{n \times m}$ with non-negligible probability, then one can solve *every* instance of certain lattice problems in the *worst case*. The most visible application of this construction is the one way hash function $h_{\mathbf{M}}(t)$ defined as

$$h_{\mathbf{M}}(t) = \mathbf{M}t \bmod q.$$

An important property of Ajtai's reduction is that in order to generate a worst case instance, one only needs to choose a matrix at random; that is, Ajtai's construction is hard on average. To emphasize this property let's consider the RSA system: the trapdoor in the RSA system is based on the fact that only for properly selected p and q there is no known polynomial time algorithm to factorize $n = pq$ without knowing p or q but if for example p and q are close to each other and $p - 1$ and $q - 1$ have small Greatest Common Divisor then the system becomes vulnerable [25]. This means that RSA system is based on a trapdoor function that is hard in worst case and needs generating hard instances for the system to work securely which in turn needs additional time and cost. This is not the case for systems based on lattice problems.

Though they are easy to implement (for example the above hash function only needs additions and multiplication modulo q), cryptographic systems that are built based on the hardness of problems on general lattices and have provable security, lack efficiency; which is because of large key space necessary to store $n \times n$ matrices and related arithmetic operations. But if lattices could have some kind of structure that made them be represented by less number of bits, efficiency of lattice-based cryptographic systems could be enhanced, provided that such structures do not affect the hardness of the underlying problems. This approach have first been used in NTRU public key cryptosystem which needs $O(n^2)$ operations to encrypt or decrypt a message of length n which is considerably faster than RSA (RSA needs $O(n^3)$ to encrypt or decrypt a message of length n)[6]. NTRU is constructed on the fact that finding short vectors in lattices is hard but there is no proof for its security.

A major improvement in efficiency of lattice-based cryptography with provable security was due to Lyubashevsky and Micciancio's worst case to average case reduction for ideal lattices [13]; as where key size and complexity decreases from $\tilde{O}(n^2)$ for general lattice to $\tilde{O}(n)$ for ideal lattices. The cost that arises because of the efficiency of public key cryptgraphic systems that are built on ideal lattices is that the underlying problem is hard to solve for this certain type of lattices; but even with their algebraic structures, still there is no known algorithm to solve hard lattice problems for these lattices more efficiently in contrast to general lattices.

In this thesis we study digital signature schemes that are based on the hardness of solving certain lattice problems. We study them in two categories: 1) signature schemes with provable security and 2) schemes that do not have a security proof. For the first category, we study a scheme proposed by Lyubashevsky and Micciancio in [14] which is an efficient lattice based digital signature with provable security based on hardness of solving approximate Shortest Vector Problem on ideal lattices. For the second category, we study GGH signature and NTRUSign. NTRUSign, is a signature scheme proposed by Hofstein *et al.*, which is built based on hardness approximating the Closest Vector Problem over ideal lattices. NTRUSign has a very similar design technique to its predecessor, GGH signature scheme, proposed by Goldreich *et al.*, but it is more efficient since it uses compact NTRU lattices.

They were first Gentry and Szydlo to observe that GGH and NTRUSign do not have zero knowledge property [4] and leak some information about the secret value in each signature. Using this information leakage, Nguyen and Regev proposed a cryptanalysis technique that recovers the secret value using 90000 signatures [19]. We study this cryptanalysis technique and implement it on a lattice of dimension 15.

Rest of this thesis is organized as follows: chapter two is a survey on lattice theory and successive minima, computational problems on lattice and basis reduction algorithms. In chapter 3 we study three signature schemes along with cryptnalaysis of two of them and we implement the attack on a small dimension lattice using MATLAB and we conclude on chapter 4.

CHAPTER 2

Lattice Theory

Throughout this thesis we will denote vectors by bold letters; a vector of vectors will be denoted by a bold letter with hat, i.e. if $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^m$ then $\widehat{\mathbf{v}} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$. By $\langle \mathbf{v}, \mathbf{u} \rangle$ we mean the inner product of $\mathbf{v}$ and $\mathbf{u}$.

By $||\mathbf{v}||_p$ we mean the l_p norm defined as

$$||\mathbf{v}||_p = \left(\sum_{i=1}^{m} v_i^p \right)^{\frac{1}{p}}$$

where $p \geq 1$ and $\mathbf{v} = \{v_1, v_2, \ldots, v_m\} \in \mathbb{R}^m$.

The l_2 norm is the Euclidean norm which will be denoted by $||\mathbf{v}||$ for any $\mathbf{v}$. When $p = \infty$ we have $||\mathbf{v}||_\infty = \max_{1 \leq i \leq m} \{v_i\}$. For any set of vectors $\mathbf{S}$, $||\mathbf{S}||$ is the length of the longest vector in $\mathbf{S}$, i.e. $||\mathbf{S}|| = \max_i ||\mathbf{s}_i||$. By $\lfloor v \rceil$ we mean the closest integer to v.

$\mathbb{Z}[x]$ is the set of the polynomial with integer coefficients. Any polynomial $f = f_0 + f_1 x^1 + \ldots + f_{n-1} x^{n-1}$ can be represented with an n-dimensional vector $(f_0, f_1, \ldots, f_{n-1})$. For any f in $\mathbb{Z}$, $\langle f \rangle$ is the set of all multiples of f. $\mathbb{Z}[x]/\langle f \rangle$ is quotient ring of $\mathbb{Z}[x]$ by $\langle f \rangle$.

We will use conventional big O notation:

- $f = O(g)$ if there is a real number c such that $f(n) \leq c \cdot g(n)$ as n goes to infinity
  - $g = \Omega(f)$ if $f = O(g)$
- $f = o(g)$ if for every $\delta > 0$, $f(n) \leq \delta \cdot g(n)$ as n goes to infinity
  - $g = \omega(f)$ if $f = o(g)$
- $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$.

2.1 Lattices

Definition 2.1. An m-dimensional *lattice* Λ is an additive discrete subgroup of $\mathbb{R}^m$. Formally, Λ is the set of all integer combinations

$$\left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}$$

of n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \in \mathbb{R}^m$ $(n \leq m)$. The set $\{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$ is called *basis* of the lattice and can be represented by the matrix

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}.$$

Using this notation the lattice can also be represented as $\{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$ where $\mathbf{Bx}$ is matrix-vector multiplication. The lattice generated by $\mathbf{B}$ is denoted by $\mathcal{L}(\mathbf{B})$. The integer n is called the *rank* of the lattice and if $n = m$ the lattice is called *full-rank*. In other words, $\mathcal{L}(\mathbf{B}) \in \mathbb{R}^m$ is full rank if and only if span of its basis vectors is equal to $\mathbb{R}^m$ [17]. If $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$, then $\mathcal{L}(\mathbf{B})$ is called integer lattice. Unless stated otherwise, we will deal with full rank integer lattices.

Two basis $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{m \times n}$ generate the same lattice if there is an unimodular matrix (a matrix which its determinant is equal to ± 1) $\mathbf{U}$ such that $\mathbf{B} = \mathbf{B}'\mathbf{U}$. So one can immediately conclude that there exist infinitely many basis for every lattice; but (as we will see later) the important issue is they can have different attributes from a cryptographic perspective. Basis with short and almost orthogonal vectors have properties that lead to solving problems that are defined on lattices and so different representations of a lattice basis can cause different behaviors and results when dealing with lattice problems. The process of achieving basis with short and almost orthogonal vectors from a basis that does not have these properties, roughly speaking, is called basis reduction.

We stated that there are many ways to represent a basis (a matrix, generally); one of them is by using Hermit Normal form of a basis.

Definition 2.2. An invertible square $n \times n$ integer matrix $\mathbf{M} = (m_{i,j})$ is in *Hermit Normal Form, HNF*, if:

- it is upper triangular,
- for all $i = 1, \dots, n,\ m_{i,i} > 0$,
- for all $i < j,\ m_{i,i} \geq m_{i,j}$.

Definition 2.3. For any basis $\mathbf{B}$ *Fundamental Parallelpiped* is defined as

$$\mathcal{P}(\mathbf{B}) = \{\mathbf{Bx} : 0 \leq x_i < 1\}.$$

One can easily see that the fundamental parallelpiped does not contain any lattice vector other than origin for any basis. Volume of the fundamental parallelepiped is a constant value for all different basis of a same lattice. This leads to an important definition, called the determinant of a lattice.

Definition 2.4. *Determinant* of a lattice $\mathcal{L}(\mathbf{B})$, denoted by $\det(\mathcal{L}(\mathbf{B}))$ is the n-dimensional volume of the fundamental parallelepiped $\mathcal{P}(\mathbf{B})$.

Determinants of different basis of the same lattice are the same; i.e. for different basis $\mathbf{B}$ and $\mathbf{B}'$ we have $\det(\mathbf{B}) = \det(\mathbf{B}')$ because $\mathbf{B} = \mathbf{B}'\mathbf{U}$ for unimodular $\mathbf{U}$. So n-dimensional volume of the fundamental parallelepiped and thus determinant of a lattice does not depend on choice of the basis.

Definition 2.5. *Gram-Schmidt orthogonalization* is a process that given any set of n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$, outputs n linearly independent mutually orthogonal vectors $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*$.

Each $\mathbf{b}_i^*$ is calculated as

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$$

and $\mu_{i,j}$'s with $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}$ are called *Gram-Schmidt coefficients*.

The following properties exist for $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ and their corresponding Gram-Schmidt orthogonalized vectors $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*$

- $\operatorname{span}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \operatorname{span}(\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*)$,
- $\{\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*\}$ is not necessarily a basis for $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ but the determinant of the lattice is equal to the product of the length of the related orthogonalized vectors of its basis, that is,

$$\det(\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)) = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|,$$

- For every i, $\mathbf{b}_i^*$ is the component of $\mathbf{b}_i$ orthogonal to $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{i-1}$,
- If ϕ_i is the angle between $\mathbf{b}_i$ and $\operatorname{span}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{i-1})$ then

$$\|\mathbf{b}_i^*\| = \|\mathbf{b}_i\| \cos(\phi_i).$$

Definition 2.6. The *projection operation* π_i is defined as

$$\pi_i(\mathbf{v}) = \sum_{j=i}^{n} \frac{\langle \mathbf{v} \cdot \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^* \cdot \mathbf{b}_j^* \rangle} \mathbf{b}_j^*.$$

We have $\mathbf{b}_i^* = \pi_i(\mathbf{b}_i)$.

Gram-Schmidt process has many applications; an important one of which is in the basis reduction algorithms (see section 2.6).

Definition 2.7. For any full rank lattice Λ, there is a *dual* lattice, denoted as Λ^* which is defined as

$$\Lambda^* = \{y \in \mathbb{R}^n | \forall x \in \Lambda, \langle x.y \rangle \in \mathbb{Z}\}.$$

One can easily verify that $(\Lambda^*)^* = \Lambda$ and $\mathcal{L}(\mathbf{B})^* = \mathcal{L}((\mathbf{B}^{-1})^T)$ [21] (where $\mathbf{B}^{-1}$ is the inverse of $\mathbf{B}$) and thus $\det(\mathcal{L}(\mathbf{B})) = \det(\mathcal{L}(\mathbf{B})^*)^{-1}$.

Definition 2.8. An integer lattice Λ is a *q-ary* lattice if $q\mathbb{Z}^n \subseteq \Lambda$ for some integer q.

Given an $n \times m$ matrix $\mathbf{C} \in \mathbb{Z}_q$, two q-ary lattices can be defined; first one is the lattice generated by the rows of $\mathbf{C}$, defined as

$$\Lambda_q(\mathbf{C}) = \left\{\mathbf{v} \in \mathbb{Z}^m : \mathbf{v} = \mathbf{C}^T\mathbf{r} \bmod q \text{ for some } \mathbf{r} \in \mathbb{Z}^n\right\},$$

and second one is the lattice that contains all the vectors that are perpendicular to all rows of $\mathbf{C}$, defined as

$$\Lambda_q^{\perp}(\mathbf{C}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{C}\mathbf{v} = 0 \bmod q\}.$$

Ajtai in [1] have proved that solving some hard problems on average for q-ary lattices can lead to solving hard problems for general lattices in worst case (see section 2.7). This proof is considered as a main step toward lattice-based cryptographic constructions.

2.2 Successive Minima

Length (with $\|\cdot\|_2$ norm) of the shortest vector, denoted by $\lambda_1(\Lambda)$, is an important constant of every lattice Λ, specially from a cryptographic point of view. One reason for this is that there is no known efficient algorithm to be able to find the shortest vector of a lattice. Indeed, finding the shortest vector can be considered as a trapdoor for cryptographic constructions. We now give a definition, called successive minima, which is the generalization of length of the shortest vector in lattices.

Definition 2.9. For any lattice Λ of rank n, the length of ith shortest vector, denoted by $\lambda_i(\Lambda)$ for $i = 1, \ldots, n$ are called the lattice's ith *successive minima*, which alternatively can be defined as

$$\lambda_i(\Lambda) = \inf\{r : \dim(\text{span}(\Lambda \cap \mathcal{B}(\mathbf{0}, r))) \geq i\}$$

where $\mathcal{B}(\mathbf{0}, r) = \{x \in \mathbb{R}^n : \|x\| \leq r\}$ is n-dimensional, **0** centered, closed ball of radius r.

Successive minima does not always form a basis for the lattice though its span equals to span of the basis. Moreover, there always exist lattice vectors whose lengths are equal to lattice's successive minima; consider the following theorem.

Theorem 2.1. *[17, Theorem 1.2] For any rank n lattice with successive minima λ_i for $i = 1, \ldots, n$ there exist lattice vectors v_i that $\|v_i\| = \lambda_i$ for $i = 1, \ldots, n$.*

We now mention two theorems that provide bounds on successive minima. These theorems, known as *Minkowski's first and second theorem*, state that $\sqrt{n}\text{det}(\Lambda)^{\frac{1}{n}}$ is an upper bound both for first successive minima and geometric mean of all successive minima of lattice Λ.

Theorem 2.2. ***Minkowski's first Theorem:*** *Let Λ be a lattice of rank n and let λ_1 be its first successive minima, then*

$$\lambda_1 < \sqrt{n}det(\Lambda)^{\frac{1}{n}}$$

Theorem 2.3. ***Minkowski's second Theorem:*** *Let Λ be a lattice of rank n and let λ_i for $i = 1, \dots, n$ be its n successive minima, then*

$$\left(\prod_{i=1}^{n} \lambda_i\right)^{\frac{1}{n}} < \sqrt{n}det(\Lambda)^{\frac{1}{n}}$$

Definition 2.10. *Hermite's Constant* of dimension n, denoted by γ_n, is supremum of ${\lambda_1}^2/\text{det}(\Lambda)^{2/n}$ over all n dimension lattices.

The exact value of the Hermite's constant is known for lattices of dimension $1 \leq n \leq 8$ [15], see table 2.1. Using the Hermite's constant, Minkowski's second theorem is as following.

Theorem 2.4. *[15, Theorem 2.6.8]: Let Λ be a lattice of rank n and let λ_i for $i = 1, \dots, n$ be its n successive minima, then for $1 \leq d \leq n$*

$$\prod_{i=1}^{d} \lambda_i \leq {\gamma_n}^d det(\Lambda)^{\frac{d}{n}}.$$

dimension n	1	2	3	4	5	6	7	8
γ_n	1	$\frac{2}{3^{1/2}}$	$\frac{2}{4^{1/3}}$	$2^{1/2}$	$2^{3/5}$	$\frac{2}{3^{1/6}}$	$2^{6/7}$	2

Table 2.1: Hermite's constant for lattices with dimension $1 \leq n \leq 8$

2.3 Cyclic and Ideal Lattices

Definition 2.11. For a vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, its *cyclic rotation*, denoted by $\text{rot}(\mathbf{x})$, is defined as

$$\text{rot}(\mathbf{x}) = \text{rot}\left((x_1, x_2, \dots, x_n)^T\right) = (x_n, x_1, \dots, x_{n-1})^T$$

and its *circulant matrix* is defined as

$$\text{Rot}(\mathbf{x}) = \left[\mathbf{x}, \text{rot}(\mathbf{x}), \text{rot}^2(\mathbf{x}), \dots, \text{rot}^{n-1}(\mathbf{x})\right]$$

In [18] a *cyclic lattice* is defined as a lattice $\mathcal{L}(\mathbf{B})$ that if a vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ then $\text{rot}(\mathbf{x}) \in \mathcal{L}(\mathbf{B})$.

Note that only one n-dimensional vector is needed to represent a cyclic lattice. This is a considerable improvement compared to $n \times n$ matrices that are necessary for representing basis of lattices that do not have any special structure. The results of using the structured cyclic lattices are that the possible compact representation reduces the necessary storage space (for the key) as well as the computational complexity of related arithmetic operations making them efficient compared to general lattices. The following lemma states that given a lattice $\mathcal{L}(\mathbf{S})$ it is possible to efficiently generate a full rank cyclic lattice.

Lemma 2.5. *[18, Lemma 3.1] There exist a polynomial time algorithm that given a full rank n-dimensional lattice in polynomial time $\mathcal{L}(\boldsymbol{B})$ outputs a vector $\boldsymbol{c} \in \mathcal{L}(\boldsymbol{B})$ that $\|\boldsymbol{c}\|_1 \leq 2n\|\boldsymbol{B}\|$ and Rot($\boldsymbol{c}$) is full rank.*

We now introduce an interesting type of structured lattices, called ideal lattices.

Definition 2.12. An *ideal lattice* is an integer lattice $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^n$ such that $\mathcal{L}(\mathbf{B}) = \{g \bmod f : g \in I\}$ for some monic polynomial $f \in \mathbb{Z}[x]$ of degree n and ideal $I \subseteq \mathbb{Z}[x]/\langle f \rangle$ [13].

When $f = x^n - 1$ one can see that $\mathcal{L}(\mathbf{B}) = \{g \bmod x^n - 1 : g \in I\}$ are cyclic lattices for ideals $I \subseteq \mathbb{Z}[x]/\langle x^n - 1\rangle$. While general integer lattices correspond to additive subgroups of $\mathbb{Z}^n$, ideal lattices correspond to ideals in the quotient rings $\mathbb{Z}[x]/\langle f \rangle$.

Consider the ideal lattice $\mathcal{L}(\mathbf{B}) = \{g \bmod f : g \in I\}$ and the ideal $I \subseteq \mathbb{Z}[x]/\langle f \rangle$; a vector is in the lattice if and only if its corresponding polynomial is in the ideal, that is

$$(f_0, f_1, \ldots, f_{n-1}) \in \mathcal{L}(\mathbf{B}) \Leftrightarrow f_0 + f_1 x + \ldots + f_{n-1}x^{n-1} \in I.$$

The following lemma states an important property of ideal lattices.

Lemma 2.6. *[13, Lemma 3.2] Let f be a monic irreducible polynomial of degree n, then every lattice that corresponds to an ideal in the quotient ring $\mathbb{Z}[x]/\langle f \rangle$ is a full rank lattice of dimension n.*

2.4 Computational Problems

There are several problems related to lattice; for some of them there exist polynomial time algorithms (see [17] section 2.2) but for the problems of the interest of this study, there are no known polynomial time algorithms that can solve them. In this section we will give definitions of three of these hard problems as well as their approximate versions [17].

Definition 2.13 (*Shortest Vector Problem, SVP*)**.** Given a basis $\mathbf{B}$, find a nonzero lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x}\| \leq \|\mathbf{y}\|$ for any other nonzero lattice vector $\mathbf{y} \in \mathcal{L}(\mathbf{B})$.

Definition 2.14 (*Closest Vector Problem, CVP*)**.** Given a basis $\mathbf{B}$, and a vector $\mathbf{t} \in \mathbb{R}^n$ find a lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x} - \mathbf{t}\| \leq \|\mathbf{y} - \mathbf{t}\|$ for any other lattice vector $\mathbf{y} \in \mathcal{L}(\mathbf{B})$.

There are no known efficient algorithms for solving SVP and CVP thus approximate versions of these problems are also considered; approximate variation of SVP asks for a lattice point which its length is within a factor of length of the shortest vector of the lattice Λ, $\lambda_1(\Lambda)$ and approximate version of CVP, given a target point $\mathbf{t}$, asks for a lattice point that is its length from $\mathbf{t}$ is at most γ time bigger than length of closest lattice point to $\mathbf{t}$ from $\mathbf{t}$.

Definition 2.15 (*Approximate SVP*)**.** Given a basis $\mathbf{B}$ and an approximation factor γ, find a nonzero lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x}\| \leq \gamma\|\mathbf{y}\|$ for any other nonzero lattice vector $\mathbf{y} \in \mathcal{L}(\mathbf{B})$.

Definition 2.16 (*Approximate CVP*)**.** Given a basis $\mathbf{B}$, a vector $\mathbf{t} \in \mathbb{R}^n$ and an approximation factor γ, find a lattice vector $x \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x} - \mathbf{t}\| \leq \gamma\|\mathbf{y} - \mathbf{t}\|$ for any other lattice vector $\mathbf{y} \in \mathcal{L}(\mathbf{B})$.

Another important hard lattice problem, for a lattice of dimension n, asks for n linearly independent lattice vectors whose lengths are at most equal to length of lattice's successive minima. The formal definition of Shortest Independent Vectors Problem and its approximate version are as follows.

Definition 2.17 (*Shortest Independent Vectors Problem, SIVP*)**.** Given a basis $\mathbf{B}$, find n linearly independent lattice vectors $S \subset \mathcal{L}(\mathbf{B})$ such that $\|S\| \leq \lambda_n$.

Definition 2.18 (*Approximate SIVP*)**.** Given a basis $\mathbf{B}$ and an approximation factor γ, find n linearly independent lattice vectors $S \subset \mathcal{L}(\mathbf{B})$ such that $\|S\| \leq \gamma \cdot \lambda_n$.

Both for SVP and CVP one can consider *Decision Problem, Optimization Problem* and *Search Problem.* SVP decision problem is to decide if there exists a vector in a lattice whose length is less than a given rational positive value, optimization problem is to find the length of the shortest vector and search problem is to find the shortest vector of a lattice. The Search Problem is harder than Optimization Problem and the Optimization Problem is harder than Decision Problem.

We now define promise problems of approximate versions of SVP and CVP and next, using these definitions, we will state three theorems on NP-hardness of SVP and CVP.

Definition 2.19 (*The promise problem GapSVP$_\gamma$*)**.** Given a lattice $\mathcal{L}(\mathbf{B})$ and a target r, decide if $\mathcal{L}(\mathbf{B})$ has a vector shorter than r (Yes instance) or does not have a vector shorter than $\gamma \cdot r$ (No instance) where γ, the gap function, is a function of the rank of the lattice; formally GapSVP$_\gamma$ is defined as follows:

- Yes instances are the pairs $(\mathbf{B}, r)$ where $\|\mathbf{x}\| \leq r$ for some nonzero lattice vector $x \in \mathcal{L}(\mathbf{B})$
- No instances are the the pairs $(\mathbf{B}, r)$ where $\|\mathbf{x}\| > \gamma \cdot r$ for all nonzero lattice vectors $\mathbf{x} \in \mathcal{L}(\mathbf{B})$

Definition 2.20 (*The promise problem GapCVP$_\gamma$*)**.** Given a lattice $\mathcal{L}(\mathbf{B})$ and a target r and a vector $\mathbf{t} \in \mathbb{R}^m$, where γ, the gap function, is a function of the rank of the lattice, GapCVP$_\gamma$ is defined as follows:

- Yes instances are the triples $(\mathbf{B}, \mathbf{t}, r)$ where $\|\mathbf{x} - \mathbf{t}\| \leq r$ for some lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$
- No instances are the the triples $(\mathbf{B}, \mathbf{t}, r)$ where $\|\mathbf{x} - \mathbf{t}\| > \gamma \cdot r$ for all lattice vectors $\mathbf{x} \in \mathcal{L}(\mathbf{B})$

Theorem 2.7. *[17, Theorem 3.1] For any $p \geq 1$, GapCVP$_1$ in the l_p norm is NP-Complete.*

Theorem 2.8. *[17, Theorem 3.2] GapSVP$_1$ in the l_∞ norm is NP-Complete.*

Theorem 2.9. *[16, Theorem 1] For any l_p norm and for any constant $\gamma \in [1, \sqrt[p]{2}]$ GapSVP$_\gamma$ is hard for NP under RUR-reductions with inverse polynomial error probability.*

2.4.1 Babai's Rounding Off Algorithm

For Closest Vector Problem, when a good basis with short vectors $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n]$ is available, one can approximate the solution for the problem using a method called *Babai's Rounding Off* algorithm [2]. Babai in [2] also introduces another method called *Nearest Plane* algorithm but for this study, we will only consider the Rounding Off algorithm.

Suppose one wishes to find a close vector to a target point t; then one needs to

- find β_i's in the following equation

$$\mathbf{t} = \sum_{i=1}^{n} \beta_i \, \mathbf{r}_i;$$

- find the closest integers α_i to each β_i and compute the vector

$$\mathbf{v} = \sum_{i=1}^{n} \alpha_i \, \mathbf{r}_i.$$

v is a close vector to **t**. Rounding Off algorithm is used in the signature schemes GGH signature and NTRUSign (sections 3.3 and 3.4).

2.5 Basis Reduction

As mentioned before, there exist infinitely many basis for every lattice. Among the many possible basis of a lattice, those which are shorter in length, or tend to be orthogonal are of special interest. In other words, for example, if one has access to a basis **R** which its vectors are close in length to the n shortest linearly independent vectors of a lattice of dimension n, then one immediately can solve approximate SVP and SIVP but if instead one only has access to $\mathbf{B} = \mathbf{R}\mathbf{U}_1\mathbf{U}_2\ldots$ where $\mathbf{U}_i$'s are unimodular matrices, then these problems cannot be solved efficiently. In this section we will study properties of different basis as well as algorithms that can generate basis that make hard problems of lattices be solved much easily.

Consider lattices of dimension 2. Let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$ and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2]$ be two different basis for a 2-dimensional lattice where vector of $\mathbf{V}$ have smaller length and suppose α and β be the angle between the vectors of $\mathbf{V}$ and $\mathbf{U}$ respectively. Since $\det(\mathcal{L})$ is constant for any choice of the basis, we will have $\alpha < \beta$. This can be easily seen in figure 2.1. Basically, basis that their vectors are short and have mutual angels of near to orthogonal are considered as *reduced basis*. Finding such a basis is the aim of every lattice reduction algorithm.

An important quantity directly related to the mutual angle of the vectors of a basis, introduced by Schnorr, is called *orthogonality defect*.

Definition 2.21. *Orthogonality defect* of basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ is defined as

$$\text{orthogonality-defect}(\mathbf{B}) = \prod_i \frac{\|b_i\|}{\|b^*{}_i\|} = \frac{\prod_i \|b_i\|}{\det(\mathbf{B})}.$$

and the *dual orthogonality defect* is defined as

$$\text{dual-orthogonality-defect}(\mathbf{B}) = \det(\mathbf{B}) \prod_i \|b_i'\|.$$

where $\mathbf{b}_i'$'s are vectors of $\mathbf{B}^{-1}$.

Since $\|\mathbf{b}_i^*\| = \|\mathbf{b}_i\| \cos(\phi_i)$ where ϕ_i is the angle between $\mathbf{b}_i$ and $\text{span}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{i-1})$ we have

$$\prod_i \frac{\|b_i\|}{\|b^*{}_i\|} = \prod_i \frac{\|b_i\|}{\|b^*{}_i\| \cos(\phi_i)} = \frac{1}{\prod_i \cos(\phi_i)} \geq 1$$

So orthogonality defect is always larger or equal to 1; the equality occurs when all the ϕ_i's are $\frac{\pi}{2}$ and when $\phi_i \neq \frac{\pi}{2}$ the orthogonality effect is larger than 1.

Lets again consider lattices of dimension 2. For a 2-dimensional lattice the concept of reduced basis is very simple; indeed a fundamental parallelepiped which its diagonals are at least as long as its edges [17] (See Figure 2.2) is geometrical interpretation of a reduced basis for 2-dimensional lattice, that is a basis $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2)$ is reduced if

$$\|\mathbf{v}_1\|, \|\mathbf{v}_2\| \leq \|\mathbf{v}_1 + \mathbf{v}_2\|, \|\mathbf{v}_1 - \mathbf{v}_2\|.$$

Indeed these conditions can be achieved in polynomial time; that is, there is polynomial time algorithm that given any basis outputs a reduced basis for any lattice of dimension 2. We will now state a theorem on two dimensional reduced basis.

Theorem 2.10. *Let $\mathcal{L}$ be a lattice with reduced basis $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2)$, then $\lambda_1 = \|\boldsymbol{v}_1\|$ and $\lambda_2 = \|\boldsymbol{v}_2\|$.*

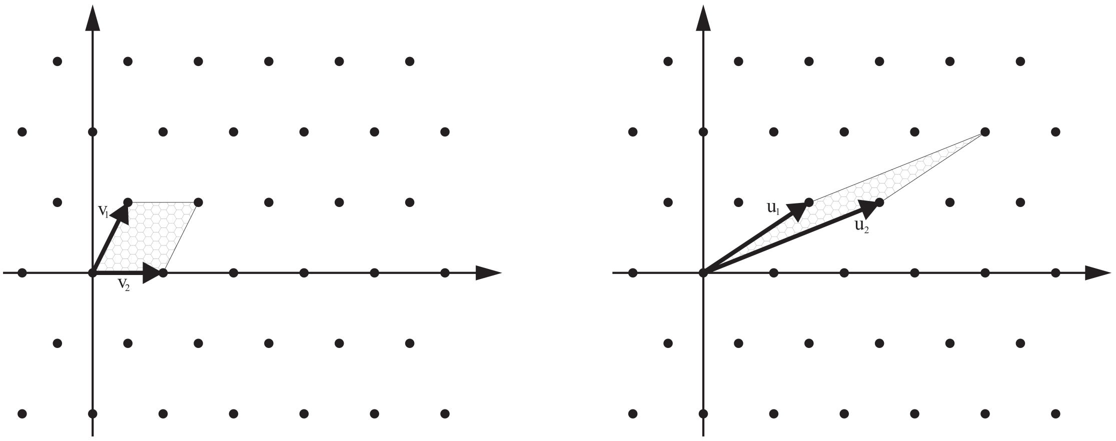

Figure 2.1: different basis

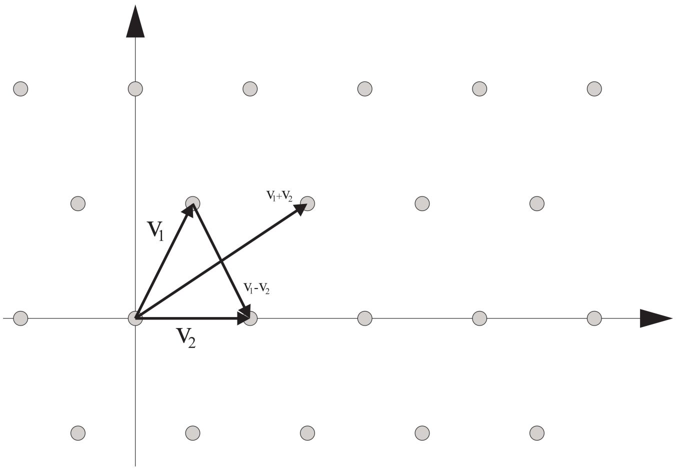

Figure 2.2: 2 dimensional reduced basis

There are different definitions (or criteria) that a lattice can be considered reduced with respect to each of them. We will now mention some of these criteria.

Definition 2.22. Let $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ be a lattice basis and $\mathbf{B}^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*]$ be its Gram-Schmidt orthogonalization. $\mathbf{B}$ is a *size reduced basis* (defined by Lagrange) if it satisfies the inequality

$$|\mu_{i,j}| \leq \frac{1}{2}, \text{ for } 1 \leq j < i \leq n.$$

One of the most important basis reduction algorithms is the one called Lenstra–Lenstra–Lovász (LLL for short) basis reduction algorithm. LLL basis reduction algorithm approximates SVP by a factor of $2^{(n-1)/2}$ in polynomial time.

Definition 2.23. A basis $\mathbf{B}$ is *LLL-reduced* [10] if it is size reduced and

$$\|\mathbf{b}_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*\|^2 \geq \frac{3}{4}\|\mathbf{b}_{i-1}^*\|^2, \text{ for } i = 2, \ldots, n.$$

Theorem 2.11. *[10, propositions 1.6, 1.11, 1.12] Let* $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ *be a LLL-reduced basis for a lattice* Λ *and* $\boldsymbol{B}^* = [\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \ldots, \boldsymbol{b}_n^*]$ *be as*

$$\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^*$$

and let $x_1, x_2, \ldots, x_t \in \mathcal{L}(\boldsymbol{B})$ *be linearly independent, then*

1. $\|b_j\|^2 \leqslant 2^{i-1} \cdot \|b^*{}_i\|^2$ *for* $1 \leq j \leq i \leq n$*,*
2. *det*$(\Lambda) \leq \prod_{i=1}^{n} \|b_i\|^2 \leq 2^{\frac{n(n-1)}{4}} \cdot$ *det*(Λ)*;*
    - *det*$(\Lambda) \leq \prod_{i=1}^{n} \|b_i\|^2$ *is called Hadamard's inequality*
3. $\|b_1\| \leq 2^{\frac{(n-1)}{4}} \cdot$ *det*$(\Lambda)^{\frac{1}{n}}$*,*
4. $\|b_1\|^2 \leqslant 2^{n-1} \cdot \|x\|^2$ *for every* $x \in \Lambda$*,*
5. $\|b_j\|^2 \leqslant 2^{n-1} \cdot$ *max* $\left\{\|x_1\|^2, \|x_2\|^2, \ldots, \|x_t\|^2\right\}$ *for* $j = 1, 2, \ldots, t$*.*

Definition 2.24. (as in [8]) A basis $\mathbf{B}$ is *Korkine-Zolotareff reduced* (KZ reduced in short) if it is size reduced and

$$\|\mathbf{b}_i^*\| = \lambda_i, \text{ for } i = 1, \ldots, n.$$

Definition 2.25. A basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ of the lattice Λ is β*-BKZ* reduced, i.e. *block Korkine-Zolotareff* reduced [22], if it is size reduced and

$$\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_{n-\beta+1})$$

are KZ reduced basis for all $i = 1, \ldots, n - \beta + 1$ (π is the projection operation; see definition 2.6).

Definition 2.26. A basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m\beta}]$ of the lattice Λ is *block* 2β*-reduced* [22] if and if it is size reduced and

$$\pi_{i\beta+1}(\mathbf{b}_{i\beta+1}), \pi_{i\beta+1}(\mathbf{b}_{i\beta+2}), \ldots, \pi_{i\beta+1}(\mathbf{b}_{(i+2)\beta})$$

are KZ reduced basis for all $i = 1, \ldots, m - 2$.

Notice that every 2β-BKZ is block 2β-reduced and every block 2-reduced basis is LLL reduced [22].

There is no known polynomial time algorithm to output a β-BKZ reduced or block 2β-reduced basis given a lattice basis, which leads to following definitions of reduced basis, semi β-BKZ reduced and semi block 2β-reduced basis.

Before stating the definitions of semi β-BKZ reduced and semi block 2β-reduced basis, let δ_β be as

$$\delta_\beta = \max \frac{\|\mathbf{b}_1\|^2}{\|\mathbf{b}_\beta^*\|^2}$$

where maximum is taken over all KZ-reduced basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\beta]$ of rank β lattices and let α_β be as

$$\alpha_\beta = \max \left(\frac{\|\mathbf{b}_1^*\| \ldots \|\mathbf{b}_\beta^*\|}{\|\mathbf{b}_{\beta+1}^*\| \ldots \|\mathbf{b}_{2\beta}^*\|} \right)$$

where maximum is taken over all KZ-reduced basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{2\beta}]$ of rank 2β lattices. Schnorr shows that $\delta_\beta \leq \beta^{1+\ln}(\beta)$ [22, corollary 2.5] and $\alpha_\beta \leq 4\beta^2$ [22, Theorem 2.7]

Definition 2.27. For a lattice basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m\beta}]$ let C_i be as

$$C_i = \prod_{j=1}^{\beta} \|\mathbf{b}_{i\beta+j}^*\|^2, \quad \text{for } i = 1, \ldots, m-1.$$

and consider the following properties

1. $C_i \leq \frac{4}{3}\alpha_\beta^\beta \, C_{i+1}$
2. $\|\mathbf{b}_{i\beta}^*\|^2 \leq 2\|\mathbf{b}_{i\beta+1}^*\|^2$
3. the k-blocks $\pi_{i\beta+1}(\mathbf{b}_{i\beta+1})$ are KZ-reduced for $i = 1, \ldots, m-1$

then the basis $\mathbf{B}$ is *semi block 2β-reduced* if the properties (1) to (3) hold and it is *semi β-BKZ reduced* if the properties (2) and (3) hold.

Note that every block 2β-reduced basis is semi block 2β-reduced and every β-BKZ reduced basis is β-BKZ reduced [22].

2.6 Basis Reduction Algorithms

It is conjectured that there is no polynomial time algorithm to approximate SVP, CVP or SIVP to a polynomial factor; a conjecture that security of most of the lattice based cryptographic functions rely on. But each of the discussed reduced basis provides upper bounds and approximations for the successive minima of a lattice. In this section we will review the bounds that algorithms which generate these reduced basis provide as well as time complexity of them.

2.6.1 Approximation Factors of Successive Minima

LLL-reduced basis provides a reasonable approximation for the successive minima:

Theorem 2.12. *[10] Let $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ be a LLL-reduced basis for a lattice Λ, then*

$$2^{1-i} \leq \frac{\|b_i\|^2}{\lambda_i} \leq 2^{n-1} \textit{ for } 1 \leq i \leq n$$

So using LLL-algorithm, one can approximate the shortest vector as well as other $n-1$ successive minima in lattice Λ by the factor of $2^{(n-1)/2}$. The following theorem gives a tighter approximation for successive minima of a lattice using KZ reduced basis:

Theorem 2.13. *[9, Theorem 2.1] Let $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ be a Korkine-Zolotareff reduced basis, then*

$$\frac{4}{i+3}(\lambda_i)^2 \leq \|\boldsymbol{b}_i\|^2 \leq \frac{i+3}{4}(\lambda_i)^2 \textit{ for } i = 1, \ldots, n.$$

Schnorr [23], using *block Korkine-Zolotareff* reduced basis gives even tighter bounds for successive minima. γ_β is Hermite's constant of lattices of dimension β (see definition 2.10).

Theorem 2.14. *[23, Theorem 3] Let $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ be a β-BKZ reduced basis, then*

$$\frac{\|\boldsymbol{b}_i\|^2}{(\lambda_i)^2} \leq \gamma_\beta^{2\frac{m-1}{\beta-1}} \frac{i+3}{4} \textit{ for } i = 1, \ldots, n.$$

Theorem 2.15. *[23, Theorem 4] Let $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ be a β-BKZ reduced basis, then*

$$\frac{(\lambda_i)^2}{\|\boldsymbol{b}_i\|^2} \leq \gamma_\beta^{2\frac{i-1}{\beta-1}} \frac{i+3}{4} \textit{ for } i = 1, \ldots, n.$$

So by last two theorems we have

$$\frac{4}{i+3}\gamma_\beta^{-2\frac{i-1}{\beta-1}} \leq \frac{\|\mathbf{b}_i\|^2}{(\lambda_i)^2} \leq \gamma_\beta^{2\frac{n-1}{\beta-1}} \frac{i+3}{4} \text{ for } i = 1, \ldots, n.$$

Theorem 2.16. *[22, Theorem 2.6] Every block 2β-reduced basis $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{m\beta}]$ of lattice Λ satisfies*

$$\|\boldsymbol{b}_1\|^2 \leq \gamma_\beta\, \alpha_\beta^{m-1} \lambda_1^2.$$

Since there are no known polynomial time algorithms that could generate β-BKZ reduced basis and block 2β-reduced basis given an arbitrary basis of a lattice, the bounds of last three theorems can not be achieved efficiently; so consider the following theorem that states the bounds that are achievable in polynomial time by a semi block 2β-reduced basis.

Theorem 2.17. *[22, Theorem 3.1] Every semi block 2β-reduced basis $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{m\beta}]$ of lattice Λ satisfies*

$$\|\boldsymbol{b}_1\|^2 \leq 2\gamma_\beta\, \delta_\beta\, \alpha_\beta^{m-2} \lambda_1^2.$$

2.6.2 Time Complexity of Basis Reduction Algorithms

Th following three theorems state the time complexity of algorithms that given an arbitrary basis of a lattice generate their LLL-reduced, KZ reduced and semi BKZ reduced basis.

Theorem 2.18. *[10, Theorem 1.26] Let Λ be an integer lattice with basis $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n]$ and let $2 \leq B \in \mathbb{R}$ be such that $\|\boldsymbol{b}_i\|^2 \leq B$, then LLL basis reduction algorithm (described in [5]) needs*

$$O(n^4 logB)$$

arithmetic operations on $O(nlogB)$ bit integers.

Theorem 2.19. *[22, Theorem 4.3] Let Λ be an integer lattice with basis $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n] \in \mathbb{Z}^{m \times n}$ with $m = O(n)$ such that $\|\boldsymbol{b}_i\|^2 \leq B$, the KZ basis reduction algorithm (algorithm C in [22]) needs*

$$n^{\sqrt{n+o(n)}} + O(n^4 logB)$$

arithmetic operations on $O(nlogB)$ bit integers.

Theorem 2.20. *[22, Theorem 3.2] Let Λ be an integer lattice with basis $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n] \in \mathbb{Z}^{m \times n}$ with $n = m\beta$ and $m = O(n)$ such that $\|\boldsymbol{b}_i\|^2 \leq B$, then both semi block 2β-reduction and semi β-BKZ reduction algorithms (algorithm A for semi block 2β-reduction and algorithm B for semi β-BKZ reduction in [22]) need*

$$O\left(n^2 \left(\sqrt{\beta^{\beta+o(\beta)} + n^2}\right) logB\right)$$

arithmetic operations on $O(nlogB)$ bit integers.

2.7 Ajtai's Reduction

For $\mathbf{M} \in \mathbb{Z}_q^{n \times m}$ consider the following q-ary lattice (see definition 2.8)

$$\Lambda_q^{\perp}(\mathbf{M}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{M}\mathbf{v} = 0 \bmod q\}$$

and let n, m and q be such that $q < m < \frac{q}{2n^4}$. Ajtai in [13] proved that being able to find a short vector in $\Lambda_q^{\perp}(\mathbf{M})$ where $\mathbf{M}$ is selected uniformly at random from $\mathbb{Z}_q^{n \times m}$ means being able to solve SVP and CVP for all lattices. In other words, if an adversary can find $\mathbf{x}$ with $\|\mathbf{x}\| < n$ such that

$$\mathbf{M}\mathbf{x} \equiv 0 \bmod q$$

for random $\mathbf{M}$ in polynomial time, then it can solve SVP and CVP for all lattices in polynomial time. This means that solving $\mathbf{M}\mathbf{x} \equiv 0 \bmod q$ is hard on *average* if SVP (or CVP) is hard in *worst case*.

An immediate result of this worst case to average case reduction is construction of collusion resistance hash functions defined as follows

$$h_{\mathbf{M}}(\mathbf{m}) = \mathbf{M}\mathbf{m} \bmod q\ .$$

Such functions are collusion resistant because a collusion leads to finding a short vector in $\Lambda_q^{\perp}(\mathbf{M})$; that is

$$h_{\mathbf{M}}(\mathbf{m}) = h_{\mathbf{M}}(\mathbf{m}') \Leftrightarrow \mathbf{M}\mathbf{m} = \mathbf{M}\mathbf{m}' \bmod q \Leftrightarrow \mathbf{M}(\mathbf{m} - \mathbf{m}') \equiv 0 \bmod q$$

and $\mathbf{m} - \mathbf{m}'$ is a short vector in $\Lambda_q^{\perp}(\mathbf{M})$.

CHAPTER 3

Lattice Based Digital Signatures

3.1 Digital Signatures

Digital signatures are one of the most important cryptographic primitive which provide authentication data integrity and non-repudiation. In other words, digital signature of a digital document

- allows the sender of the document to introduce himself and allows receiver to verify if the sender of the document is actually who he or she claims to be,
- guarantees that the transmitted data is a accurate and not altered,
- provides proof so that the sender can not deny having sent the document.

A digital signature scheme generally has three algorithms; key generation, signing and verification algorithm:

- **Key generation algorithm**, G, gets the security parameter of the scheme as input and generates a pair of keys (s, p) called secret (private) key and public key. Secret key will be used for signing the document and is only known to the legitimate signer and public key as its name implies is know to everyone who wishes to communicate with the signer and is used to verify the signature;
- **Signing algorithm**, S, gets the message m that will be signed and the secret key and gives the signature of the message $\mathsf{S}_s(m)$. Only the legitimate signer can do this because the secret key is only known to him;
- **Verification algorithm**, V_p, gets the signature of the message, the message itself and the public key and verifies the signature and outputs 1 if it is correct and 0 otherwise.

Formally, a digital signature scheme can be defined as follows:

Definition 3.1. A *digital signature scheme* is a 3-tuple of polynomial time algorithms $(\mathsf{G}, \mathsf{S}, \mathsf{V})$ that

$$\mathsf{Pr}[\mathsf{V}_p(m, \mathsf{S}_s(m)) = 1] = 1$$

The security of every digital signature scheme is based on the hardness of solving a specific problem when only restricted information is available; that is, for every signature scheme a problem or so called a trapdoor is needed that can be solved efficiently only if certain information about certain factors of problem is available. Integer Factorization Problem and Discrete Logarithms problem are two well studied such problems that are being used in digital signature schemes. In case of the lattice based digital signature schemes the problem can be Shortest Vector Problem or Closest Vector Problem which (as we saw in section 2.4) are hard to solve and even approximate without having almost orthogonal basis with short vectors.

3.2 An efficient one-time lattice based digital signature with provable security

The signature scheme that will be studied in this section is constructed by Lyubashevsky and Micciancio [14] and is based on hardness of approximating the shortest vector in structured cyclic or ideal lattices. In section 2.3 we saw that cyclic lattices need less space and time for the arithmetic operation that they get involved.

In this signature scheme, a hash function family $\mathcal{H}_{R,m}$ will be used which is defined as follows: let R be the ring $\mathbb{Z}[x]/\langle f\rangle$ where f is an irreducible polynomial in $\mathbb{Z}[x]$. Each hash function from $\mathcal{H}_{R,m}$ has an element $\hat{\mathbf{e}} \in R^m$ and maps elements of R^m to R, more precisely

$$\mathcal{H}_{R,m} = \{h_{\hat{\mathbf{e}}} : \hat{\mathbf{e}} \in R^m\}, \text{ where } h_{\hat{\mathbf{e}}}(\hat{\mathbf{a}}) = \hat{\mathbf{e}} \odot \hat{\mathbf{a}} \text{ for } \hat{\mathbf{a}} \in R^m$$

where the dot product $\odot$ is defined as $\hat{\mathbf{e}} \odot \hat{\mathbf{a}} = (\mathbf{e}_1\mathbf{a}_1 + \mathbf{e}_2\mathbf{a}_2 + \ldots + \mathbf{e}_m\mathbf{a}_m)$.

In the terms of security and efficiency, this signature scheme can be briefly described as follows: the signature of a message of n bits length is of length $\tilde{O}(k)$ which can be signed and verified in $\tilde{O}(k) + \tilde{O}(n)$ time complexity and assuming that $\tilde{O}(k^2)$-SVP is hard in all ideal lattices, this scheme is strongly unforgeable in chosen message attack model [14, Theorem 1].

An important factor in the efficiency of this signature scheme is that in contrast to the one-way function based signature schemes that sign only one bit at a time and need to call the one-way function n time for an n bit message, it only uses the one-way function twice for every signature.

To sign a message, one picks at random two inputs $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ and compute their hashes using the lattice based hash function h. The signature of the message $\mathbf{m}$ will be the linear combination of $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, that is, $\hat{\mathbf{s}} = \hat{\mathbf{a}} \cdot \mathbf{m} + \hat{\mathbf{b}}$ and since the lattice based hash function is homomorphic, signature can easily be verified as

$$h(\hat{\mathbf{a}} \cdot \mathbf{m} + \hat{\mathbf{b}}) = h(\hat{\mathbf{a}}) \cdot \mathbf{m} + h(\hat{\mathbf{b}})$$

We will now give the formal explanation of the key generation, signing and verification algorithms of the one-time signature.

3.2.1 Key Generation

Key Generation Algorithm

Input: irreducible polynomial $f \in \mathbb{Z}[x]$ of degree n

Output: signing key $(\hat{\mathbf{a}}, \hat{\mathbf{b}})$ and verification key $(h, h(\hat{\mathbf{a}}), h(\hat{\mathbf{b}}))$

1. Set $p = (\phi n)^3$, $m = \lceil \log n \rceil$ and $R = \mathbb{Z}_p[x]/f$,
2. For all positive i's define sets A_i and B_i as
$$A_i = \left\{ \hat{\mathbf{y}} \in R^m \text{ such that } \|\hat{\mathbf{y}}\|_\infty \leq 5ip^{1/m} \right\},$$
$$B_i = \left\{ \hat{\mathbf{y}} \in R^m \text{ such that } \|\hat{\mathbf{y}}\|_\infty \leq 5\phi nip^{1/m} \right\},$$
3. Choose $h \in \mathcal{H}_{R,m}$ uniformly at random,
4. Pick a string $r \in \{0,1\}^{\lfloor \log^2 n \rfloor}$ uniformly at random,
5. **if** $r = 0^{\lfloor \log^2 n \rfloor}$ **then** set $j = \lfloor \log^2 n \rfloor$ **else** set j to the position of the first 1 in the string r,
6. Pick $\hat{\mathbf{a}} \in A_j$ and $\hat{\mathbf{b}} \in B_j$ uniformly at random,
7. Give $(\hat{\mathbf{a}}, \hat{\mathbf{b}})$ as signing key and $(h, h(\hat{\mathbf{a}}), h(\hat{\mathbf{b}}))$ as verification key.

For the ring R, the parameter ϕ is defined as

$$\phi(R) = \min \{ j : \forall\, \mathbf{a}, \mathbf{b} \in R,\ \|\mathbf{ab}\|_\infty \leq jn\|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty \}.$$

The motivation for this definition is that upper bounds for $\|\mathbf{a}+\mathbf{b}\|_\infty$ and $\|a\mathbf{b}\|_\infty$ for integer a can be obtained easily but for upper bounding $\|\mathbf{ab}\|_\infty$ with $\mathbf{a}, \mathbf{b} \in R$ one needs to take into account the possible raise in the coefficients of **ab** when it is reduced modulo R. Suppose $\mathbf{a}, \mathbf{b} \in \mathbb{Z}[x]$ with degree $< n$. Then the degree of **ab** at most can be $2n-2$ and its coefficient can be at most $n\|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty$ but when reduced modulo f its maximum coefficient can be larger than $n\|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty$. In other words, for maximum coefficient of product of two elements in the ring $R = \mathbb{Z}[x]/f$ after reduction modulo f the following holds

$$(\phi(R) - 1)\, n\|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty < \|\mathbf{ab}\|_\infty \leq \phi(R) n\|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty.$$

We will see later (theorem 3.1) that the the approximation factor of SVP over R, on which the security of this signature scheme depends on, is determined by $\phi(R)$ which in turn is determined by polynomial f. So it is logical to keep $\phi(R)$ as small as possible (which in turn will keep the approximation factor of SVP smaller which in turn makes the problem harder); in his Ph.D. thesis [12], Lyubashevsky shows that $\phi(x^n - 1) = \phi(x^n + 1) = 1$ and $\phi(x^n + x^{n-1} + \ldots + 1) = 2$ and also provides bounds on the value of $\phi(R)$.

Notice that the keys $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ are selected at random but not uniformly; that is, keys with smaller coefficient are more probable to be selected, here is why: for $1 \leq j \leq \lfloor \log^2 n \rfloor$ each $\hat{\mathbf{a}}$ is selected uniformly at random from the set A_j with probability 2^{-j} and for $j = \lfloor \log^2 n \rfloor$ each $\hat{\mathbf{a}}$ is selected uniformly at random from the set A_j with probability 2^{-j+1}. Since $\hat{\mathbf{a}}$ is selected uniformly at random from the set $A_1 \subset A_2 \subset \ldots \subset A_{\lfloor \log^2 n \rfloor}$, $\hat{\mathbf{a}}$'s are indeed chosen from the set $A_{\lfloor \log^2 n \rfloor}$ at random but not uniformly. The same argument holds for $\hat{\mathbf{b}}$.

3.2.2 Signing and Verification

Signing Algorithm

Input: $(\hat{\mathbf{a}}, \hat{\mathbf{b}})$ and message $\mathbf{m} \in R$ with $\|\mathbf{m}\|_\infty \leq 1$

Output: signature $(\hat{\mathbf{s}}, \mathbf{m})$

1. $\hat{\mathbf{s}} = \hat{\mathbf{a}} \cdot \mathbf{m} + \hat{\mathbf{b}}$.

Verification Algorithm

Input: message $\mathbf{m}$, signature $\hat{\mathbf{s}}$ and verification key $(h, h(\hat{\mathbf{a}}), h(\hat{\mathbf{b}}))$

Output: signature verification response

1. **if** $\|\hat{\mathbf{s}}\|_\infty \leq 10\phi p^{1/m} n \log^2 n$ and $h(\hat{\mathbf{s}}) = h(\hat{\mathbf{a}}) \cdot \mathbf{m} + h(\hat{\mathbf{b}})$ give ”accept” **else** give ”reject”.

The lattice-based hash function h is homomrphic so $h(\hat{\mathbf{a}} \cdot \mathbf{m} + \hat{\mathbf{b}}) = h(\hat{\mathbf{a}}) \cdot \mathbf{m} + h(\hat{\mathbf{b}})$ and we have

$$\|\hat{\mathbf{a}}\|_\infty \leq 5p^{1/m}$$

$$\|\hat{\mathbf{b}}\|_\infty \leq 5\phi n p^{1/m}$$

since $\hat{\mathbf{a}} \in A_{\lfloor \log^2 n \rfloor}$ and $\hat{\mathbf{b}} \in B_{\lfloor \log^2 n \rfloor}$, so

$$\|\hat{\mathbf{s}}\|_\infty = \|\hat{\mathbf{a}} \cdot \mathbf{m} + \hat{\mathbf{b}}\|_\infty \leq \|\hat{\mathbf{a}} \cdot \mathbf{m}\|_\infty + \|\hat{\mathbf{b}}\|_\infty \leq \phi n \|\hat{\mathbf{a}}\|_\infty \|\mathbf{m}\|_\infty + \|\hat{\mathbf{b}}\|_\infty \leq 10\phi p^{1/m} n \log^2 n$$

and verification algorithm accepts correct signatures generated by the signing algorithm.

3.2.3 Security of the Signature Scheme

Collision problem $Col_{d,\mathcal{H}_{R,m}}(h)$ is defined as follows

Definition 3.2. Given a hash function h from the hash function family $\mathcal{H}_{R,m}$, the *collision problem* $Col_{d,\mathcal{H}_{R,m}}(h)$ ask for two elements $\hat{\mathbf{a}}$ and $\hat{\mathbf{a}'}$ from R^m with $\|\hat{\mathbf{a}'}\|_\infty, \|\hat{\mathbf{a}}\|_\infty \leq d$ where $\hat{\mathbf{a}} \neq \hat{\mathbf{a}'}$ such that $h(\hat{\mathbf{a}}) = h(\hat{\mathbf{a}'})$.

For defined parameters, being able to forge the signature of an arbitrary message after seeing a message **m** and its signature $\hat{\mathbf{s}}$ means being able to solve the collision problem over ideal lattices which in turns means being able to approximate the shortest vector in an ideal lattice with the factor of $\tilde{O}(\phi^5 n^2)$. The following two theorems prove security guarantees of the system but before stating them, we will review the parameters; R is the ring $\mathbb{Z}[x]/\langle f \rangle$ where f is an irreducible polynomial and p, m and d are as

$$p = (\phi n)^3, \quad m = \lceil \log n \rceil \text{ and } d = 10\phi p^{1/m} n \log^2 n.$$

Theorem 3.1. *[14, Theorem 6] If there exist a polynomial time algorithm that solves the collision problem $Col_{d,\mathcal{H}_{R,m}}(h)$ then there is a polynomial time algorithm that for every lattice corresponding to an ideal in R solves $\tilde{O}(\phi^5 n^2)$-SVP*

Theorem 3.2. *[14, Theorem 7] If there is a polynomial time adversary that after seeing a message **m** and its signature $\hat{\mathbf{s}}$ can correctly sign an arbitrary message **m′** with non-negligible probability, then there is a polynomial algorithm that can solve $Col_{d,\mathcal{H}_{R,m}}(h)$.*

And lastly, the following theorem states that if an adversary only have access to verification keys, the message, the hash function used in key generation algorithm and the signature, the probability that the adversary can recover the signing keys is negligible.

Theorem 3.3. *[14, Lemma 8] Let **m** be a message, $\hat{\boldsymbol{s}}$ be its signature and $(h, \boldsymbol{V}, \boldsymbol{W})$ be the verification key. Then if for any signing key $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{w}})$ with $\hat{\boldsymbol{v}} \in A_{\lfloor log^2 n-1 \rfloor}$ and $\hat{\boldsymbol{w}} \in B_{\lfloor log^2 n-1 \rfloor}$ that the followings hold*

$$h(\hat{\boldsymbol{v}}) = \boldsymbol{V}$$
$$h(\hat{\boldsymbol{w}}) = \boldsymbol{W}$$
$$\hat{\boldsymbol{s}} = \hat{\boldsymbol{v}} \cdot \boldsymbol{m} + \hat{\boldsymbol{w}}$$

*then with non-negligible probability, $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{w}})$ is not the actual key that has been used to sign the message **m**.*

3.3 GGH Signature Scheme

GGH signature scheme, proposed by Goldreich, Goldwasser, and Halevi [1], is based on the hardness of approximating the Closest Vector Problem. The secret information is a good basis for a lattice $\mathcal{L}$ while the public information of the signature scheme is bad basis of the same lattice. After its introduction, the GGH signature did not gather the necessary interest but it sure has significant influence on the design of the later signatures schemes, such as NTRUSign.

3.3.1 Key Generation

The private key of the signature scheme is a good basis, with short almost orthogonal (with low dual orthogonality defect) vector of a full ranked integer lattice $\mathcal{L}$. Dimension of the lattice $\mathcal{L}$ should be set considering the fact that for a lattice of dimension n, the necessary space for key storage as well and running time of the algorithm will vary by a factor of $O(n^2)$. For dimensions of between 60-80 Golgreich *et al.* at [5] stated that they had found basis with very small orthogonality defect in their experiments but conjecture that dimensions between 250-300 will be proper. For generating the private key, that is the good basis **R**, two distributions are proposed in [5]:

- Choosing **R** at random where it is uniformly distributed over $\{-k, \ldots, k\}^{n \times n}$. The value of k has almost no effect on the quality of the generated basis and thus it is preferred to work with (small) integers between -4 and 4.
- Choose the noise matrix $\mathbf{R}'$ at random where it is uniformly distributed over $\{-k, \ldots, k\}^{n \times n}$ and add it to the box $r \cdot I_n \in \mathbb{R}^n$ to get the private basis; that is $\mathbf{R} = \mathbf{R}' + r \cdot I_n$. Larger r's lead to **R**'s with smaller dual orthogonality defect. Based on their experiments, Golgreich *et al.* state that the best value for r is about $\sqrt{n} \cdot k$.

Figure 3.1 shows an example for a dimension 15 secret basis **R**, generated using the first distribution (for the scheme to be secure against lattice reduction algorithm, dimension need to be more than 200).

$$\begin{bmatrix}
-1 & 4 & -4 & 1 & -3 & 1 & 1 & -2 & 0 & 3 & 0 & -4 & 2 & 4 & 4 \\
-3 & -4 & 2 & -4 & 1 & -4 & -1 & -1 & 3 & 2 & 0 & -1 & 4 & 1 & 4 \\
1 & 2 & 4 & 1 & -4 & 1 & 4 & -4 & 2 & -2 & -4 & 4 & 0 & 0 & 2 \\
3 & -4 & 2 & -3 & 1 & 1 & 2 & 0 & 0 & -4 & 0 & -4 & 2 & -3 & 1 \\
-4 & -4 & -1 & -4 & 3 & -4 & 2 & 2 & 0 & 1 & 2 & 4 & 2 & 2 & -2 \\
4 & -2 & -3 & -2 & 2 & 1 & 3 & 3 & -3 & -4 & -3 & 0 & -3 & 4 & -3 \\
-4 & 2 & -2 & -4 & 4 & 0 & 4 & -3 & 2 & -3 & 2 & 3 & -1 & 1 & -1 \\
0 & 4 & 1 & 0 & -1 & 0 & -4 & -3 & 1 & -4 & 3 & -4 & 1 & 4 & 3 \\
4 & -3 & -1 & 0 & 4 & -3 & 1 & 3 & 0 & -3 & 1 & 2 & -4 & 2 & -4 \\
-2 & -1 & -2 & -3 & 0 & 0 & 4 & -4 & -4 & 3 & -4 & -1 & 4 & 1 & -3 \\
-1 & -3 & 2 & -2 & -4 & -4 & 1 & 0 & -4 & -1 & 2 & 1 & 4 & 4 & -2 \\
2 & -1 & 2 & 0 & 3 & 3 & -3 & 1 & -4 & 2 & 2 & 4 & 4 & 4 & 1 \\
1 & -2 & -1 & 0 & -3 & 1 & 3 & -1 & 4 & -3 & -4 & -2 & -4 & 2 & 1 \\
-4 & -1 & -3 & 1 & -4 & 1 & 1 & 1 & 4 & 3 & 2 & 2 & -3 & 2 & -4 \\
-2 & 3 & 4 & 3 & 2 & -1 & -3 & 3 & -1 & 4 & -4 & 3 & -4 & -1 & 0
\end{bmatrix}$$

Figure 3.1: Secret basis **R** chosen from uniform distribution over $\{-4, \ldots, 4\}^{15 \times 15}$

After generating the private basis **R**, which has short and almost orthogonal vectors for the lattice $\mathcal{L}$, the public basis is generated for the same lattice from **R**. In [5], two methods for generating public basis **B** are proposed:

- Choose the vectors of **R** one by one and add linear combinations of the other vectors (with coefficients from the set $\{-1, 0, 1\}$, where the coefficient 0 is selected with more probability so that the values of the basis B does not increase harshly) to each selected vector. $2n$ mixing steps prevent LLL algorithm from reducing **R**.

- Choose unimodular matrices $\mathbf{U}_i$ with $\mathbf{U}_i = \mathbf{U}_{1_i}\mathbf{U}_{2_i}$ where $\mathbf{U}_{1_i}$'s are lower triangular and $\mathbf{U}_{2_i}$'s are upper triangular matrices, both with diagonal entries of ± 1 and other entries form the set $\{-1, 0, 1\}$ and generated **B** as

  $$\mathbf{B} = \mathbf{R} \cdot \mathbf{U}_1 \cdot \mathbf{U}_2 \cdot \ldots$$

  At least four $\mathbf{U}_i$'s should be used in the above multiplication so that **R** can be calculated from **B** using LLL algorithm.

From these two methods, authors proposed the first one because based on their experiments, the second method generate lattice with larger entries. Figure 3.2 shows the public basis corresponding the secret basis in figure 3.1. **B** is computed by multiplying **R** to unimodular matrices.

$$\begin{bmatrix}
55 & -19 & 87 & 60 & -35 & 27 & 54 & 115 & 15 & 93 & -18 & -47 & 2 & 37 & 45 \\
-37 & 5 & -26 & -47 & 11 & -25 & -77 & -12 & -46 & -10 & -45 & 60 & 35 & 0 & -80 \\
48 & 57 & -1 & -19 & 48 & 18 & 2 & 52 & -49 & 26 & -62 & 8 & -24 & -6 & -40 \\
-7 & -16 & -37 & 3 & -15 & -9 & -19 & -31 & -39 & 62 & 34 & -14 & -25 & -20 & -11 \\
-80 & -24 & -16 & 4 & -25 & -13 & -35 & -64 & 33 & -82 & 20 & 39 & 31 & 20 & -46 \\
-24 & -38 & 31 & 52 & -53 & 62 & 31 & 33 & -12 & 20 & 61 & -32 & -54 & 30 & 10 \\
-93 & -41 & 26 & 121 & -117 & 83 & 34 & -54 & 80 & -5 & 79 & -68 & -55 & 108 & -51 \\
-99 & -55 & -11 & 3 & -93 & 29 & -53 & -21 & -40 & 16 & 40 & -4 & -17 & 60 & -48 \\
-69 & -28 & -29 & -14 & -29 & 10 & -28 & -61 & 1 & -64 & 49 & 14 & -10 & 10 & -11 \\
37 & -12 & 49 & 96 & -58 & 46 & 66 & 60 & 13 & 123 & 23 & -91 & -48 & 48 & 20 \\
-83 & -45 & -30 & -54 & -29 & -30 & -71 & -29 & -48 & -57 & 20 & 57 & 34 & -2 & -21 \\
-26 & 38 & -70 & -45 & 29 & -117 & -60 & -38 & -48 & 13 & -85 & 54 & 80 & -24 & -83 \\
12 & -38 & 69 & 32 & -28 & 83 & 19 & 86 & -18 & 30 & 32 & -7 & -46 & 22 & 15 \\
-17 & -80 & 99 & 26 & -32 & 33 & 17 & 48 & 84 & -76 & 47 & 32 & 43 & 16 & 75 \\
47 & 89 & -16 & -52 & 96 & -15 & 8 & 14 & -3 & -76 & -89 & 39 & 15 & -26 & -28
\end{bmatrix}$$

Figure 3.2: Public basis **B** corresponding to secret basis in figure 3.1

3.3.2 Signing and Verification

The private key of the signature scheme is a good basis with short vector and small orthogonality defect for the lattice $\mathcal{L}$. One having such a lattice basis **R** can solve CVP easily; a problem that is hard for any adversary which only has the basis **B**. This fact is the trapdoor of the GGH signature scheme.

The message is mapped to point $\mathbf{m} \in \mathbb{R}^n$ and then a close vector to **m** is found using Babai's rounding off algorithm (see section 2.4.1); this lattice point is the signature. Verification consists of checking if the signature is a lattice point and if the distance between the message digest and the signature is small. We now explain these steps in detail and to clarify more, we give numerical examples for each step.

Signing Algorithm

Input: private key $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n]$ message digest $\mathbf{m} \in \mathbb{R}^n$

Output: signature $(\mathbf{s}, \mathbf{m})$

1. Find α_i's such that $\mathbf{m} = \sum_{i=1}^{n} \alpha_i \mathbf{r}_i$,
2. **For** $i = 1, \ldots, n$ set $\beta_i = \lfloor \alpha_i \rceil$,
3. Set $\mathbf{s} = \sum_{i=1}^{n} \beta_i \mathbf{r}_i$,
4. Give $(\mathbf{s}, \mathbf{m})$.

We now give a numerical example: let **m** (which for simplicity has been chosen uniformly random from $\{-100, \ldots, 100\}^{n \times n}$) be the message hash to be signed:

$$\mathbf{m}^t = \begin{bmatrix} -63 & -26 & 25 & 56 & -84 & 86 & 55 & -3 & -13 & -11 & -39 & 2 & 2 & 64 & 59 \end{bmatrix}$$

we should find the coefficients α_i's that give exact value of **m** in terms of vectors of **R**; if $\mathbf{a}^t = [\alpha_1, \alpha_2, \ldots, \alpha_{15}]$, we will have

$$\mathbf{m} = \mathbf{Ra} \;\; \rightarrow \begin{bmatrix} -63 \\ -26 \\ 25 \\ 56 \\ -84 \\ 86 \\ 55 \\ -3 \\ -13 \\ -11 \\ -39 \\ 2 \\ 2 \\ 64 \\ 59 \end{bmatrix} = \mathbf{R} \begin{bmatrix} 38.2502 \\ 26.3372 \\ 22.1608 \\ -94.3819 \\ -26.9109 \\ 15.0387 \\ -10.8858 \\ -2.7378 \\ -3.1256 \\ 38.6152 \\ 16.0900 \\ 2.5185 \\ -47.1654 \\ -5.96474 \\ -6.2541 \end{bmatrix}$$

to generate the signature of the message digest **m**, we round the coefficients α_i's to their nearest integer and multiply to the secret basis **R** to find a lattice point **s** in the lattice $\mathcal{L}(\mathbf{R})$. In other words, the signature of message digest **m** will be:

$$\mathbf{R}\lfloor\mathbf{a}\rceil = \mathbf{s} \;\; \rightarrow \;\; \mathbf{R}\begin{bmatrix} 38 \\ 26 \\ 22 \\ -94 \\ -27 \\ 15 \\ -11 \\ -3 \\ -3 \\ 39 \\ 16 \\ 3 \\ -47 \\ -6 \\ -6 \end{bmatrix} = \begin{bmatrix} \textbf{-62} \\ \textbf{-23} \\ \textbf{27} \\ \textbf{52} \\ \textbf{-82} \\ \textbf{81} \\ \textbf{54} \\ \textbf{-6} \\ \textbf{-16} \\ \textbf{-10} \\ \textbf{-39} \\ \textbf{4} \\ \textbf{1} \\ \textbf{67} \\ \textbf{61} \end{bmatrix}$$

Verification Algorithm

Input: the signature $(\mathbf{s}, \mathbf{m})$ and public key **B**

Output: signature verification response

1. **if s** $\in \mathcal{L}$ (check using public basis **B**) and $||\mathbf{s} - \mathbf{m}|| \leq \mathcal{N}$ give "accept"; **else** give "reject".

Note that for the signature and the message we have

$$\mathbf{s} - \mathbf{m} \in \mathcal{P}_{1/2}(\mathbf{R}) = \{\mathbf{xR} \mid \mathbf{x} \in [-1/2, 1/2]^n\}.$$

This can be interpreted as follows: the signing algorithm is reduction of the message m modulo the fundamental parallelepiped generated by **R**. This property (which similarly also exist in NTRUSign) turns out to be a major weakness of the signature scheme; that is used in successful cryptanalysis of GGH signature (and NTRUSign) in [19].

In our numerical example, to verify the signature, we should first check if **s** is indeed a lattice point. We have $\mathcal{L}(\mathbf{R}) = \mathcal{L}(\mathbf{B})$, so if the solution to the equation $\mathbf{Bx} = \mathbf{s}$ is in $\mathbb{Z}^{15}$, we can be sure that **s** is a lattice point (since a lattice is combinations of its basis vectors with integer coefficients). We have:

$$\mathbf{s} = \mathbf{B} \begin{bmatrix} 6060 \\ -27320 \\ -65140 \\ 29050 \\ 6350 \\ 55010 \\ 10890 \\ -60 \\ -8760 \\ 2260 \\ -28150 \\ 15550 \\ 40280 \\ 7330 \\ 22680 \end{bmatrix}$$

so $\mathbf{s} \in \mathcal{L}(\mathbf{B})$. Now we should check if the signature **s** is close enough to the message **m**; that is check if $\mathbf{s} - \mathbf{m}$ is small:

$$\mathbf{s} - \mathbf{m} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ -4 \\ 2 \\ -5 \\ -1 \\ -3 \\ -3 \\ 1 \\ 0 \\ 2 \\ -1 \\ 3 \\ 2 \end{bmatrix}$$

and thus **s** is correct signature of **m**.

3.4 NTRUSign

NTRUSign signature scheme, introduced by Hoffstein *et al.* [7], is a lattice-based signature scheme which its security relies on the hardness of approximating the closest vector to a target point; approximate-CVP. The lattices underlying the approximate-CVP problem for NTRUSign correspond to ideals in the ring $R = \mathbb{Z}[x]/(x^n - 1)$ where its operations are addition and convolution multiplication $*$ for $f = f_0 + f_1 x^1 + \ldots +$

$f_{n-1}x^{n-1} = [f_0, f_1, \ldots, f_{n-1}]$ and $g = g_0 + g_1x^1 + \ldots + g_{n-1}x^{n-1} = [g_0, g_1, \ldots, g_{n-1}]$ defined as

$$(f * g)_k = \sum_{i=0}^{k} f_i\, g_{k-i} + \sum_{i=k+1}^{n-1} f_i\, g_{n+k-i} = \sum_{i+j \equiv k \bmod n} f_i\, g_j$$

Using matrix notation, the convolution multiplication can be stated as

$$[(f * g)_0, (f * g)_1, \ldots, (f * g)_{n-1}] = [f_0, f_1, \ldots, f_{n-1}] \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ g_{n-1} & g_0 & \cdots & g_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & \cdots & g_0 \end{bmatrix}.$$

Consider the following lattice for given integers n and q and polynomial $k \in R$:

$$\mathcal{L} = \mathcal{L}_k(n, q) = \left\{(v, v') \in R^2 \mid v \equiv v' * k \bmod q\right\}.$$

This lattice, which is called *convolution modular lattice*, is an $2n$-dimensional lattice with det$(\mathcal{L}) = q^n$. Note that $\mathcal{L}$ is a q-ary lattice and $q\mathbb{Z}^{2n} \subseteq \mathcal{L} \subseteq \mathbb{Z}^{2n}$.

Now choose two invertible f and g in R and let F and G in R be generated such that as $f * G - F * g = q$ (F and G can be generated using *resultant mapping*, for details see [7] section 4.); we will have

$$\det \begin{pmatrix} f & F \\ g & G \end{pmatrix} = q;$$

define $k = G/g = F/f \bmod q$, then

$$\begin{pmatrix} f & F \\ g & G \end{pmatrix} \text{ and } \begin{pmatrix} 1 & k \\ 0 & q \end{pmatrix}$$

both are basis for the lattice $\mathcal{L}_k(n, q)$.

3.4.1 Key Generation

We will now show how and why the NTRUSign signature scheme works. The secret key used for signing is pair of two small randomly chosen polynomials in the quotient ring $R = \mathbb{Z}[x]/(x^n - 1)$ with predefined number of coefficients equal to 1.

Key Generation Algorithm

Input: positive integers n, q, d_f, d_g

Output: Private key (f, g) and public key k.

1. Choose polynomials f with d_f ones and g with d_g ones randomly. Both f and g should be invertible modulo q. (i.e. there should be f^{-1} such that $f * f^{-1} \equiv 1 \text{mod} q$)
2. Find small F and G that $f * G - F * g = q$.
3. Set $k = F * f^{-1} = G * g^{-1} \bmod q$
4. Give the pair (f, g) as private key and k as public key

Value for the parameters d_f and d_g depends on the level of the security that will (be expected to) be provided by the system. For example at [6], for the NTRU encryption scheme, values for d_f and d_g differ for three different proposed levels of security (in [6] coefficients of f and g can also be -1). Details on method of finding the polynomials F and G are provided in [7]. Coefficients of f and g can only be 1 or 0 but F and G have larger coefficients.

The lattice $\mathcal{L} = \mathcal{L}_k(n, q) = \{(v, v') \in R^2 \mid v \equiv v' * k \bmod q\}$ defined above is called the *transpose* NTRU lattice. In standard NTRU lattice, the polynomial k, or the private key of the signature scheme, is defined as $k = f^{-1} * g \bmod q$. We only consider the signature scheme based on transposed NTRU lattice, which is also recommended in [7]. It is now worth mentioning the following theorem for standard NTRU lattice:

Theorem 3.4. *[7, Theorem 1] Let f, g, F, $G \in R$ be as*

$$f * G - F * g = q,$$

let $k = f^{-1} * g$ mod q and let $\mathcal{L}$ be the lattice generated by $\{(1, k), (0, q)\}$

1) $\{(f, g), (F, G)\}$ is also a basis for $\mathcal{L}_k(n, q)$

2) If F' and G' also satisfy

$$f * G' - F' * g = q$$

then there is an element $c \in R$ such that $F' = F + c * f$ and $G' = G + c * f$.

A hash function h is required for singing procedure which for a document D does the hashing two steps: first a hash function h_1 maps the document to a α-bit $h_1(D)$ and then a second function

$$h_2(h_1(D)) : (\mathbb{Z}/2\mathbb{Z})^\alpha \rightarrow (\mathbb{Z}/q\mathbb{Z})^n$$

is used to get the message digest $\mathbf{m} = h_2(h_1(D))$.

Let $(\mathbf{m}_1, \mathbf{m}_2) \in R^2$ and note that if $\mathbf{v}$ is close vector to $\mathbf{m}$, then $\mathbf{w} + \mathbf{v}$ is a close vector to $\mathbf{w} + \mathbf{m}$. So given $(\mathbf{m}_1, \mathbf{m}_2) \in R^2$ one only needs to find a close vector in $\mathcal{L}_k(n, q)$ to the point $(\mathbf{0}, \mathbf{m}_2 - \mathbf{m}_1 * k) \in R^2$. Thus the target poins in approximate-CVP problem in NTRUSign are all in the form $(\mathbf{0}, \mathbf{m})$.

Signing procedure will also need a norm; for $(\mathbf{u}, \mathbf{v}) \in R^2$, $\|(\mathbf{u} \bmod q, \mathbf{v} \bmod q)\|$ is defined as

$$\min_{\mathbf{k}_1, \mathbf{k}_2 \in R} \{\|(\mathbf{u} + \mathbf{k}_1 q, \mathbf{v} + \mathbf{k}_2 q)\|\}.$$

and $\|(\mathbf{u}, \mathbf{v})\| = \sqrt{\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2}$ where $\| \cdot \|$ is centered Euclidean norm; that is

$$\|\mathbf{r}\|^2 = \sum_{i=0}^{n-1} r_i^2 - (1/n) \left(\sum_{i=0}^{n-1} r_i \right)^2.$$

3.4.2 Signing

We saw that instead of finding a close vector to $(\mathbf{m}_1, \mathbf{m}_2) \in R^2$ one can find a close vector in $\mathcal{L}_k(n, q)$ to the point $(\mathbf{0}, \mathbf{m}_2 - \mathbf{m}_1 * k) \in R^2$. So a message vector $(\mathbf{m}_1, \mathbf{m}_2) \in R^2$ will be considered as $(\mathbf{0}, \mathbf{m})$ in the signature scheme. Signing a message vector $(\mathbf{0}, \mathbf{m})$ is the procedure of finding a close lattice point using Babai's rounding off procedure.

Suppose (x, y) are values of coefficients of $(\mathbf{0}, \mathbf{m})$ when it is represented as a linear combination of the basis vectors

$$\begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

that is

$$(\mathbf{0}, \mathbf{m}) = (x, y) \begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

and (x, y) will be equal to

$$(x, y) = (\mathbf{0}, \mathbf{m}) \begin{pmatrix} G & -F \\ -g & f \end{pmatrix} /q.$$

Signing Algorithm

Input: message **m** and the private key (f, g)

Output: the signature $(\mathbf{m}, \mathbf{s})$

1. Set $(x, y) = (\mathbf{0}, \mathbf{m}) \begin{pmatrix} G & -F \\ -g & f \end{pmatrix} /q = ((-\mathbf{m} * g)/q \ , \ (\mathbf{m} * f)/q)$,
2. Set $\varepsilon = \lfloor x \rceil$ and $\varepsilon' = \lfloor y \rceil$,
3. Set $\mathbf{s} = \varepsilon f + \varepsilon' g$,
4. Give $(\mathbf{m}, \mathbf{s})$ as the signature.

3.4.3 Verification

Verification Algorithm

Input: the signature $(\mathbf{m}, \mathbf{s})$ and public key k

Output: signature verification response

```
1. Set $\mathbf{t} = \mathbf{s} * k \bmod q$
2. Set $b = \|(\mathbf{s}, (\mathbf{t} - \mathbf{m}) \bmod q\|$
3. **If** $b \leq \mathcal{N}$ give "accept" **else** give "reject".
```

By the definition, one can see that

$$\|(\mathbf{s}, (\mathbf{t} - \mathbf{m}) \bmod q\| = \min_{\mathbf{k}_1, \mathbf{k}_2 \in R} \left(\|\mathbf{s} + \mathbf{k}_1 q\|^2 + \|(\mathbf{t} - \mathbf{m}) + \mathbf{k}_2 q\|^2 \right)^{1/2}.$$

There is balancing factor in β the equation

$$b = \min_{\mathbf{k}_1, \mathbf{k}_2 \in R} \left(\|\mathbf{s} + \mathbf{k}_1 q\|^2 + \|(\mathbf{t} - \mathbf{m}) + \mathbf{k}_2 q\|^2 \right)^{1/2}$$

as

$$b = \min_{\mathbf{k}_1, \mathbf{k}_2 \in R} \left(\|\mathbf{s} + \mathbf{k}_1 q\|^2 + \beta^2 \|(\mathbf{t} - \mathbf{m}) + \mathbf{k}_2 q\|^2 \right)^{1/2}$$

which is proposed to be 1 in [7]. For $\mathbf{t} = \mathbf{s} * k \bmod q$ we have:

$$\begin{aligned}
t &= \mathbf{s} * k \bmod q \\
&= (\varepsilon f + \varepsilon' g) * k \bmod q \\
&= -\left((x - \lfloor x \rceil) f + (y - \lfloor y \rceil) g\right) * k \bmod q \\
&= -\left(((-\mathbf{m} * g)/q - \lfloor (-\mathbf{m} * g)/q \rceil) f + ((\mathbf{m} * f)/q - \lfloor (\mathbf{m} * f)/q \rceil) g\right) * k \bmod q \\
&= -((-\mathbf{m} * g)/q - \lfloor (-\mathbf{m} * g)/q \rceil) f * k - ((\mathbf{m} * f)/q - \lfloor (\mathbf{m} * f)/q \rceil) g * k \bmod q \\
&= -((-\mathbf{m} * g)/q - \lfloor (-\mathbf{m} * g)/q \rceil) f * (F * f^{-1}) - ((\mathbf{m} * f)/q - \lfloor (\mathbf{m} * f)/q \rceil) g * (G * g^{-1}) \bmod q \\
&= -((-\mathbf{m} * g)/q - \lfloor (-\mathbf{m} * g)/q \rceil) * F - ((\mathbf{m} * f)/q - \lfloor (\mathbf{m} * f)/q \rceil) * G \bmod q \\
&= -((-\mathbf{m} * g * F)/q - \lfloor (-\mathbf{m} * g * F)/q \rceil) - ((\mathbf{m} * f * G)/q - \lfloor (\mathbf{m} * f * G)/q \rceil) \bmod q \\
&= (\mathbf{m} * g * F)/q + \lfloor (-\mathbf{m} * g * F)/q \rceil - (\mathbf{m} * f * G)/q + \lfloor (\mathbf{m} * f * G)/q \rceil \bmod q \\
&= \mathbf{m} * (g * F - f * G)/q + \lfloor \mathbf{m} * (-g * F + f * G)/q \rceil \bmod q \\
&= \mathbf{m} - \lfloor \mathbf{m} \rceil \bmod q
\end{aligned}$$

It is worth mentioning that the signature is indeed the pair $(\mathbf{s}, \mathbf{t})$ but since $\mathbf{t}$ can computed from $\mathbf{s}$ using k, only $\mathbf{s}$ is sent. The verification is simply verifying that the signature $(\mathbf{s}, \mathbf{t})$ is close enough to the message digest $(\mathbf{0}, \mathbf{m})$ and this is checked by comparing the distance to the norm bound $\mathcal{N}$.

3.5 Cryptanalysis of GGH signature and NTRUSign

We saw that the security of GGH signature scheme relies on the hardness of solving the approximate Closest Vector Problem over lattices. For a GGH signature to be secure, the efficiency of the systems has to decrease because of the need for lattices of high dimension. Indeed this problem also applies for any system based on hardness of lattice problems over general lattices (here by general we mean lattices that do not have any special structure or property). If a lattice basis cannot be represented by less elements, any computation related to it will need $O(n^2)$ space and time. This problem can be considered as a motivation for the introduction of structured lattices (see section 2.3), of which are NTRU lattices. NTRUSign, the signature scheme based on the hardness of approximation the closest vector in NTRU lattices, though there is no known proof of security for it, can be considered as efficient as signature schemes which are based on Integer Factorization Problem and Discrete Logarithm Problem; this is because of the the structure of ideal lattices (note that the later two classes of signature schemes also do not have any proof for their security). NTRUSign, a very special instance of GGH, uses GGH's design specification along with the compact NTRU lattices.

In this chapter, we will explain a brilliant cryptanalysis technic, exploiting the lack of zero knowledge property of the GGH signature scheme (which first was used by Gentry and Szydlo in [4]), proposed by Nguyen and Regev in [19]. The weakness of which they take advantage in their cryptanalysis is that unlike signature schemes based on IFP and DLP, every signature in the GGH signature scheme and NTRUSign leaks some information about the secret component; which in turn results to recovering the secret basis.

3.5.1 The Hidden Parallelepiped Problem

The signature of the message digest $\mathbf{m}$ in GGH signature scheme is

$$\mathbf{s} = \lfloor \mathbf{m}\mathbf{R}^{-1} \rceil \mathbf{R}$$

where $\mathbf{R}$ is the secret basis. So for correct signature, $\mathbf{s} - \mathbf{m}$ is always in the parallelepiped spanned by secret basis, that is

$$\mathbf{s} - \mathbf{m} \in \mathcal{P}_{1/2} = \{\mathbf{x}\mathbf{R} \mid \mathbf{x} \in [-1/2, 1/2]\}.$$

Considering that the parallelepiped spanned by a basis of a lattice is a fundamental domain and also that the hash function

$$h : A \to B,\ h(\mathcal{D}) = \mathbf{m}$$

generates $\mathbf{m}$'s randomly over B, it is valid to assume that for polynomially many message digests $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_k$ and their signatures $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$, the vectors $\mathbf{m}_1 - \mathbf{s}_1, \mathbf{m}_2 - \mathbf{s}_2, \ldots, \mathbf{m}_k - \mathbf{s}_k$ are independent and distributed uniformly random over $\mathcal{P}_{1/2} = \{\mathbf{x}\mathbf{R} \mid \mathbf{x} \in [-1/2, 1/2]\}$. Figures 3.3 and 3.4 show how this distribution looks like for 5000 and 10000 $\mathbf{m}_i - \mathbf{s}_i$'s respectively. The Hidden Parallelepiped Problem, seeks for approximation of the secret basis $\mathbf{R}$ given polynomial many $\mathbf{m}_i - \mathbf{s}_i$'s in the parallelepiped $\mathcal{P}(\mathbf{R})$; in more precise words:

Definition 3.3. Let $\mathbf{R} \in \mathbb{R}^{n \times n}$ be a invertible matrix, let $\mathcal{P}(\mathbf{R})$ be the parallelepiped spanned by $\mathbf{R}$:

$$\mathcal{P}(\mathbf{R}) = \{\mathbf{xR} \mid \mathbf{x} \in [-1, 1]\}$$

and let $U(\mathcal{P})$ denote the uniform distribution over the parallelepiped $\mathcal{P}$. The *Hidden Parallelepiped Problem (HPP)* seeks for good approximations of rows of $\pm\mathbf{R}$ given polynomially many samples from $U(\mathcal{P}(\mathbf{R}))$.

Considering $[-1, 1]$ rather than $[-1/2, 1/2]$ is for simplifying the calculations.

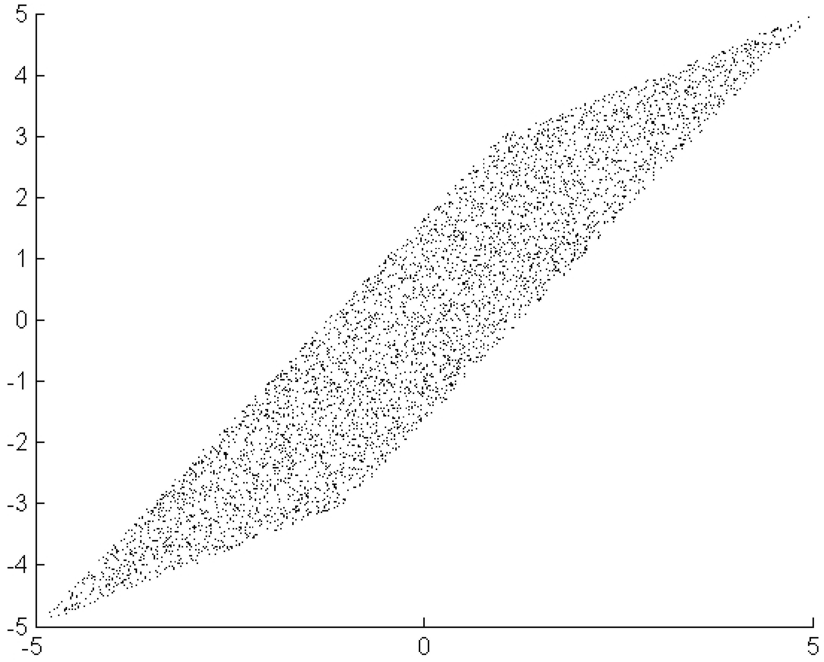

Figure 3.3: Distribution of 5000 $\mathbf{m}_i - \mathbf{s}_i$'s in $\mathcal{P}$

3.5.2 Solving HPP

The method proposed in [19] for solving HPP and recovering the vectors of the secret basis $\pm\mathbf{R}$ can be summarized in following steps. Later we will describe each step in more detail.

- Given the distribution of $\mathbf{r} = \mathbf{m}_i - \mathbf{s}_i$'s over $\mathcal{P}(\mathbf{R})$, approximate its covariance matrix of $U(\mathcal{P}(\mathbf{V}))$.
- Reduce the HPP to a problem where the parallelepiped in replaced with a hypercube (Hidden Hypercube Problem).
- Reduce the Hidden Hypercube Problem to minimizing problem of the forth moment.

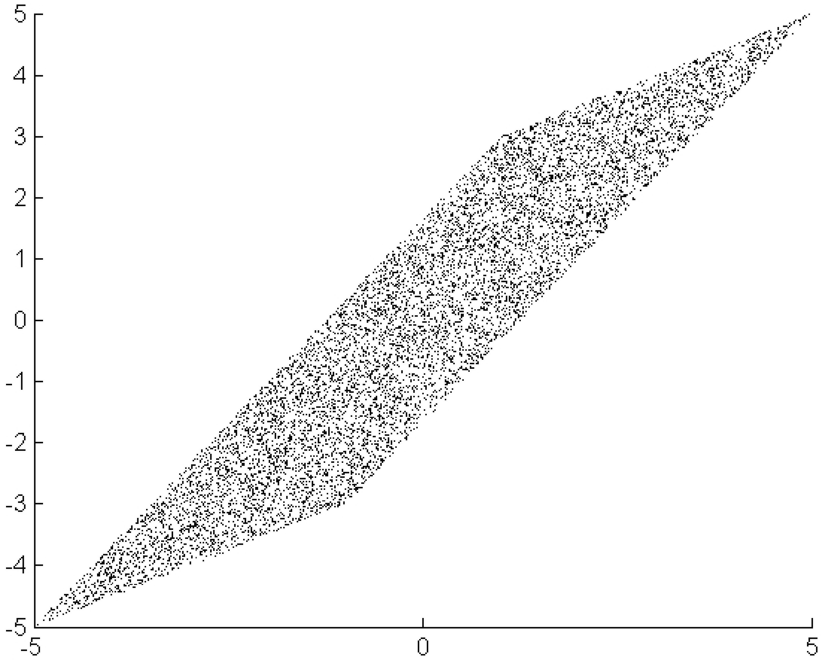

Figure 3.4: Distribution of 10000 $\mathbf{m}_i - \mathbf{s}_i$'s in $\mathcal{P}$

Covariance Matrix of $U(\mathcal{P}(\mathbf{V}))$

Each $\mathbf{r}$ from the parallelepipid $\mathcal{P}(\mathbf{R})$ can be written in the form $\mathbf{r} = \mathbf{xR}$. By the assumption that is made about the independency and uniform distribution of $\mathbf{r}$'s over $\mathcal{P}(\mathbf{R})$, one can conclude that $\mathbf{x}$'s are also independent and uniformly distributed over $[-1, 1]^n$.

For $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ the expectation of $x_i \cdot x_j$ is zero and expectation of $x_i \cdot x_i$ is $1/3$. This leads to following lemma:

Lemma 3.5. *Let $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ be an invertible matrix and let $\boldsymbol{r}$ be chosen uniformly random from $\mathcal{P}(\boldsymbol{R})$, then*

$$Exp[\boldsymbol{r}^t\boldsymbol{r}] = \boldsymbol{R}^t\boldsymbol{R}/3$$

"Exp" denotes the expected value. So using lemma 3.5., in order to approximate the Gram matrix of $\mathbf{R}^t$, where $\mathbf{R}^t\mathbf{R}/3$ is covariance matrix of $U(\mathcal{P}(\mathbf{V}))$, one can simply calculate the average of $\mathbf{r}^t\mathbf{r}$ over all $\mathbf{r}$'s from $\mathcal{P}(\mathbf{R})$ and multiply it by 3.

Reducing HPP to Hidden Hypercube Problem

The aim of this step is to map the hidden parallelepiped $\mathcal{P}(\mathbf{R})$ to a hypercube $\mathcal{P}(\mathbf{C})$ so that approximating the rows of $\pm\mathbf{C}$ enables approximating the rows of $\pm\mathbf{R}$. The following lemma shows how to do such a reduction.

Lemma 3.6. *Let $\mathbf{R} \in \mathbb{R}^{n \times n}$ be an invertible matrix, let $\mathbf{G}$ denote the symmetric positive definite matrix $\mathbf{R}^t\mathbf{R}$ and let $\mathbf{L}$ be a unique lower triangular matrix such that $\mathbf{G}^{-1} = \mathbf{L}\mathbf{L}^t$ is Cholesky factorization of $\mathbf{G}^{-1}$. Then $\mathbf{C} = \mathbf{R}\mathbf{L} \in \mathbb{R}^{n \times n}$ satisfies the followings:*

a) Rows of $\mathbf{C}$ are pairwise orthogonal unit vectors and $\mathcal{P}(\mathbf{C})$ is a unit hypercube.

b) If $\mathbf{r}$ is chosen uniformly at random from $\mathcal{P}(\mathbf{R})$, then $\mathbf{c} = \mathbf{r}\mathbf{L}$ is uniformly ditributed over $\mathcal{P}(\mathbf{C})$

So if one can approximate rows of $\pm\mathbf{C}$, then by multiplying the approximations of rows of $\pm\mathbf{C}$ by $\mathbf{L}^{-1}$, one can approximate rows of $\pm\mathbf{R}$. Figure 3.5 shows the distribution of $\mathbf{rL}$'s in $\mathcal{P}(C)$ and figure 3.6 shows the distribution of samples in $\mathcal{P}(\mathbf{R})$ and $\mathcal{P}(\mathbf{C})$ for $\mathbf{R} = \begin{bmatrix} 3 & 4 \\ -2 & -1 \end{bmatrix}$.

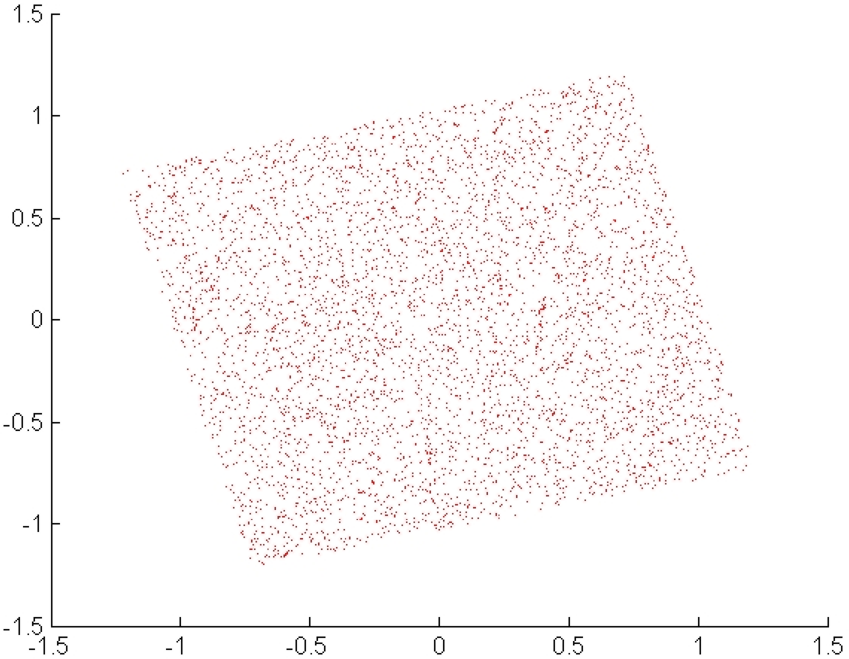

Figure 3.5: Distribution of 5000 $\mathbf{c} = \mathbf{rL}$'s in $\mathcal{P}(\mathbf{C})$

Hidden Hypercube Problem

For $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$, Nguyen and Regev in [19] define the k-th moment (which is a funcion, not a value) over a vector $\mathbf{w} \in \mathbb{R}^n$ as

$$\mathrm{mom}_{\mathbf{C},k} = \mathrm{Exp}[\langle \mathbf{c}, \mathbf{w} \rangle^k].$$

This function is used to approximate the rows of $\pm\mathbf{C}$; the following lemma says how:

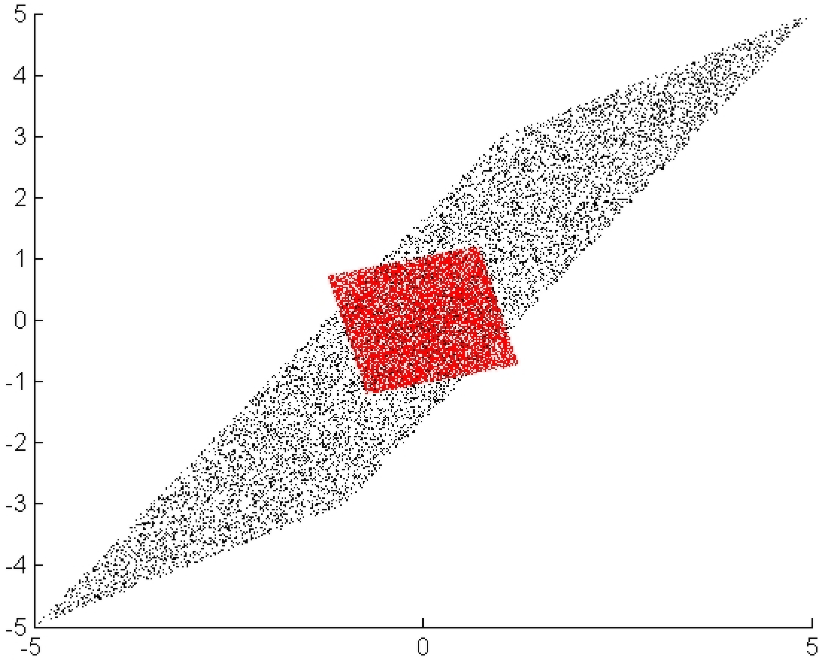

Figure 3.6: Distribution of 10000 $\mathbf{m}_i - \mathbf{s}_i$'s and $\mathbf{c} = (\mathbf{m}_i - \mathbf{s}_i)\mathbf{L}$'s

Lemma 3.7. *Let $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n]$ be orthogonal real invertible matrix. Then the global minimum of mom$_{\mathbf{C},4}$ over the unit sphere of $\mathbb{R}^n$ is 1/5 and this minimum is obtained at $\pm\boldsymbol{c}_1, \pm\boldsymbol{c}_2, \ldots, \pm\boldsymbol{c}_n$.*

So if one can find the values that global minimum of mom$_{\mathbf{C},4}$ is achieved, then one can solve the Hidden Hypercube problem; that is, one can approximate the rows of $\pm\mathbf{C}$ and once $\mathbf{C}$ is approximated, one can approximate the row of $\pm\mathbf{R}$ by multiplying $\pm\mathbf{C}$ to $\mathbf{L}^{-1}$

3.5.3 Implementation of Cryptanalysis of GGH Signature for n=15

In the last part of this study we implement the cryptanalysis technic proposed by Nguyen and Regev on GGH signature scheme for a lattice of dimension 15. We do not generate signature of message hashes and instead, considering the randomness assumption on the differences of the message hashes and their signatures over the parallelepiped, we generate random samples distributed uniformly in the fundamental parallelepiped of the secret basis $\mathbf{R}$ and use these samples to recover rows of the secret basis $\mathbf{R}$. We first generate a basis for the GGH signature scheme in dimension 15; we choose the basis uniformly random from $\{-4, \ldots, 4\}^{15\times15}$. Note that for the cryptanalysis of GGH signature and NTRUSign there is no need for the public basis of the schemes. Figure 3.7 shows the secret basis $\mathbf{R}$ that we will try to recover using sample from $\mathcal{P}(\mathbf{R})$.

$$\begin{bmatrix}
-3 & 2 & 3 & 4 & -1 & 0 & -3 & -1 & 3 & -2 & 2 & 2 & 4 & 4 & -1 \\
-2 & 2 & -1 & 1 & 1 & 0 & -4 & -4 & 4 & 4 & 1 & 1 & -4 & -2 & 0 \\
1 & 2 & 4 & 0 & -3 & 2 & 1 & -2 & 3 & -3 & 0 & 4 & 2 & -1 & 1 \\
4 & 0 & -4 & 4 & -2 & 0 & 0 & 3 & -1 & -4 & 1 & 4 & 2 & -4 & 0 \\
0 & 0 & -1 & 1 & 4 & -4 & -2 & -1 & 2 & 4 & -2 & -3 & 0 & -2 & 2 \\
0 & 1 & -2 & 3 & -4 & -3 & 1 & 3 & -2 & 4 & 3 & 1 & 4 & -1 & 2 \\
-3 & -1 & 2 & 2 & -1 & 3 & -2 & -1 & 3 & -4 & 2 & 3 & 0 & 1 & 4 \\
-3 & -3 & 1 & 0 & 0 & 4 & -1 & -2 & 2 & -1 & -3 & 4 & -3 & -4 & 4 \\
1 & 4 & 3 & 4 & 4 & 2 & -2 & 0 & 0 & 3 & 1 & -2 & -3 & 2 & -4 \\
1 & -3 & -1 & -2 & 2 & -1 & -2 & -2 & 2 & 1 & 0 & 1 & -4 & 2 & -4 \\
1 & 1 & 1 & -4 & -3 & 4 & 3 & 2 & 1 & 0 & 0 & 3 & 0 & 1 & 1 \\
2 & 2 & -4 & -2 & 1 & -4 & -1 & -1 & 1 & 1 & -4 & -1 & -2 & 2 & -1 \\
-2 & -1 & -2 & -1 & -1 & 2 & -1 & -1 & 0 & 3 & -3 & 3 & 4 & -4 & -3 \\
3 & -4 & 4 & -4 & 4 & -4 & -4 & 2 & 2 & 1 & -2 & -2 & 4 & -3 & -2 \\
-2 & -2 & 0 & -2 & -2 & -2 & 2 & -2 & -3 & -4 & -3 & -2 & 2 & 4 & 0
\end{bmatrix}$$

Figure 3.7: Secret basis **R** chosen from uniform distribution over $\{-4, \ldots, 4\}^{n \times n}$

We then generate 5000 samples (**r**'s) from the parallelepiped $\mathcal{P}(\mathbf{R})$. For dimension 2, the distribution of our samples are as figure 3.3. Figure 3.8 shows 3 of such sample in dimension 15 we will use to recover **R**.

$$\begin{bmatrix} -3.5844 & -4.1546 & -1.7225 & -2.8799 & -0.4401 & 1.5681 & -8.1336 & \ldots \end{bmatrix.$$

$$\begin{bmatrix} \ldots & -6.9550 & 9.8222 & -0.0860 & -6.8200 & 10.2191 & -1.3300 - 3.3201 & -0.0659 \end{bmatrix}$$

$$\begin{bmatrix} 2.9075 & -1.2551 & -1.0454 & 3.1229 & -1.9117 & -0.8421 & -5.1271 & \ldots \end{bmatrix.$$

$$\begin{bmatrix} \ldots & 1.7932 & 4.8183 & 3.5090 & 0.5925 & 8.3555 & 4.0423 & -6.8763 & -0.9266 \end{bmatrix}$$

$$\begin{bmatrix} 3.1066 & 0.6920 & -0.8043 & -2.4481 & -0.3353 & 0.3675 & -6.9831 & \ldots \end{bmatrix.$$

$$\begin{bmatrix} \ldots & -4.7981 & 8.8440 & -1.1781 & -5.6136 & 9.0639 & -1.2641 & -0.6672 & -5.5559 \end{bmatrix}$$

Figure 3.8: 3 examples of 5000 samples from $\mathcal{P}(\mathbf{R})$

After that we calculate the Gram matrix of $\mathbf{R}^t$; to do so, we first calculate $\mathbf{r}^t\mathbf{r}$ for all 5000 **r**'s and then compute the expected value of $\mathbf{r}^t\mathbf{r}$. Since **r**'s are distributed uniformly random in $\mathcal{P}(\mathbf{R})$, the expected value of $\mathbf{r}^t\mathbf{r}$'s is equal to their average. We then multiply the average by 3 to get Gram matrix of $\mathbf{R}^t$. Figure 3.9 shows the result.

$$\left[\begin{array}{cccccccc}
74.6554 & 2.4461 & -19.1635 & -16.6049 & 14.2425 & -34.5066 & 12.7673 & \ldots \\
2.4461 & 75.2003 & -0.7795 & 46.8357 & -12.6886 & 12.8266 & 7.0564 & \ldots \\
-19.1635 & -0.7795 & 97.5316 & -2.4039 & 15.7156 & 35.4602 & -17.9827 & \ldots \\
-16.6049 & 46.8357 & -2.4039 & 108.1403 & -11.7515 & 15.6876 & -19.1100 & \ldots \\
14.2425 & -12.6886 & 15.7156 & -11.7515 & 98.1232 & -36.0566 & -54.5877 & \ldots \\
-34.5066 & 12.8266 & 35.4602 & 15.6876 & -36.0566 & 114.5336 & 19.8070 & \ldots \\
12.7673 & 7.0564 & -17.9827 & -19.1100 & -54.5877 & 19.8070 & 75.4779 & \ldots \\
43.0597 & -0.9860 & -12.2181 & 5.8066 & -17.3730 & -13.1430 & 27.5092 & \ldots \\
-19.6802 & 3.7933 & 35.2471 & 1.8730 & 23.7297 & 18.0095 & -53.6485 & \ldots \\
4.3277 & 21.0741 & -25.7060 & -3.815 & 948.1769 & -38.9047 & -30.7182 & \ldots \\
-1.5605 & 29.5972 & 14.7583 & 53.8611 & -25.1994 & 20.2438 & 3.0068 & \ldots \\
-16.8236 & 0.0336 & 2.1753 & 21.7455 & -62.3310 & 70.2884 & 3.3894 & \ldots \\
5.9276 & -10.1776 & 12.2824 & 8.5733 & -49.0909 & -26.5903 & 14.8580 & \ldots \\
-17.2107 & 27.1830 & 22.1247 & -0.6719 & -4.4324 & -8.6303 & 16.1214 & \ldots \\
-30.1234 & -6.7631 & 4.2369 & 13.3626 & -36.0918 & 21.6731 & 17.5655 & \ldots
\end{array}\right.$$

$$\left.\begin{array}{ccccccccc}
\ldots & 43.0597 & -19.6802 & 4.3277 & -1.5605 & -16.8236 & 5.9276 & -17.2107 & -30.1234 \\
\ldots & -0.9860 & 3.7933 & 21.0741 & 29.5972 & 0.0336 & -10.1776 & 27.1830 & -6.7631 \\
\ldots & -12.2181 & 35.2471 & -25.7060 & 14.7583 & 2.1753 & 12.2824 & 22.1247 & 4.2369 \\
\ldots & 5.8066 & 1.8730 & -3.8159 & 53.8611 & 21.7455 & 8.5733 & -0.6719 & 13.3626 \\
\ldots & -17.3730 & 23.7297 & 48.1769 & -25.1994 & -62.3310 & -49.0909 & -4.4324 & -36.0918 \\
\ldots & -13.1430 & 18.0095 & -38.9047 & 20.2438 & 70.2884 & -26.5903 & -8.6303 & 21.6731 \\
\ldots & 27.5092 & -53.6485 & -30.7182 & 3.0068 & 3.3894 & 14.8580 & 16.1214 & 17.5655 \\
\ldots & 66.0444 & -38.8449 & -1.7232 & 22.1957 & -4.7590 & 44.3823 & -15.3867 & -1.0139 \\
\ldots & -38.8449 & 78.0766 & 7.9364 & 0.1650 & 33.1324 & -24.8811- & 11.7225 & 11.5231 \\
\ldots & -1.7232 & 7.9364 & 127.9151 & -5.0964 & -41.6868 & -23.6503 & -28.8657 & -31.2171 \\
\ldots & 22.1957 & 0.1650 & -5.0964 & 72.7466 & 16.7048 & 6.5053 & 17.0661 & 8.5128 \\
\ldots & -4.7590 & 33.1324 & -41.6868 & 16.7048 & 103.1964 & 15.9278 & -36.3540 & 28.3296 \\
\ldots & 44.3823 & -24.8811 & -23.6503 & 6.5053 & 15.9278 & 130.5102 & -17.5784 & 2.5296 \\
\ldots & -15.3867 & -11.7225 & -28.8657 & 17.0661 & -36.3540 & -17.5784 & 111.6769 & -21.2989 \\
\ldots & -1.0139 & 11.5231 & -31.2171 & 8.5128 & 28.3296 & 2.5296 & -21.2989 & 88.4743
\end{array}\right]$$

Figure 3.9: Approximation of $\left(\mathbf{R}^t\mathbf{R}\right)$; i.e. $3 \cdot$ Average $[\mathbf{r}^t\mathbf{r}] = \left(\mathbf{R}^t\mathbf{R}\right)$

In order to reduce the Hidden Parallelepiped Problem to Hidden Hypercube Problem, we will need Cholesky factorization of the Gram matrix of $\mathbf{r}^t\mathbf{r}$; that is we need $\mathbf{L}$ and $\mathbf{L}^{-1}$ where $\left(\mathbf{R}^t\mathbf{R}\right)^{-1} = \mathbf{L}\mathbf{L}^t$. Figure 3.10 shows the matrix $\mathbf{L}^{-1}$.

To map the parrallelepiped to the hypercube we multiply each $\mathbf{r}$ by the matrix $\mathbf{L}$ and get samples from the hypercube. We compute the forth moment

$$\text{Exp}[\langle \mathbf{c}, \mathbf{w} \rangle^4]$$

using the 5000 samples from the hypercube. Using the fmincon function from MATALB's Optimization Toolbox, we find minimums of forth moment provided that $w_1^2 + w_2^2 + \ldots + w_{15}^2 = 1$ for $\mathbf{w} = [w_1, w_2, \ldots, w_{15}]$.

When the $\mathbf{w}$'s where the forth moment has its minimum value is calculated, we multiply each $\mathbf{w}$ by the matrix $\mathbf{L}^{-1}$ to recovers the ± of rows of the secret basis. Figure 3.11 shows the results after rounding the entries to their nearest integers (since the basis $\mathbf{R}$ is integer), eliminating the repeated rows and also eliminating the rows with entries greater than 4 or less than -4.

$$\left[\begin{array}{ccccccccc}
1.1484 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\
0.7275 & 0.9663 & 0 & 0 & 0 & 0 & 0 & \ldots \\
0.7845 & -0.4875 & 4.9782 & 0 & 0 & 0 & 0 & \ldots \\
1.5282 & 6.6305 & -1.0199 & 5.7163 & 0 & 0 & 0 & \ldots \\
-0.1169 & -1.1938 & 0.3709 & 2.9401 & 1.9077 & 0 & 0 & \ldots \\
-3.3773 & 0.5121 & 5.2187 & 0.4442 & 1.5719 & 6.7567 & 0 & \ldots \\
2.1760 & 1.7491 & 1.9706 & -4.3248 & -1.9083 & 1.9850 & 4.5038 & \ldots \\
5.1695 & -0.1876 & -0.0794 & -2.4481 & 1.0406 & -0.6125 & 0.9269 & \ldots \\
-0.8924 & -0.2723 & 5.4820 & -0.1552 & 4.5897 & -1.3749 & -6.5129 & \ldots \\
-2.5912 & 2.8569 & -1.3568 & 0.0573 & -0.2804 & -1.1726 & -1.3477 & \ldots \\
1.0774 & 3.0011 & 0.9704 & 5.9711 & -0.5968 & 0.8413 & -0.5065 & \ldots \\
-1.5388 & 1.0586 & 0.6442 & 1.8940 & -5.4554 & 7.0102 & 0.2137 & \ldots \\
0.2560 & -0.5233 & 1.4022 & 0.7606 & -4.4371 & -2.4556 & 1.5570 & \ldots \\
-2.3699 & 2.4757 & 2.2422 & 0.2465 & -1.2711 & -0.3306 & 1.9715 & \ldots \\
-3.2025 & -0.7190 & 0.4504 & 1.4206 & -3.8371 & 2.3042 & 1.8675 & \ldots
\end{array}\right.$$

$$\left.\begin{array}{ccccccccc}
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 5.2583 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & -3.1357 & 7.4908 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & -1.0928 & 1.9911 & 8.7312 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 3.0920 & -0.4681 & 1.8906 & 7.9868 & 0 & 0 & 0 & 0 \\
\ldots & -1.3209 & 3.1583 & -4.2764 & 2.0076 & 9.2244 & 0 & 0 & 0 \\
\ldots & 3.7092 & -2.3567 & -2.5263 & 0.8234 & 0.9214 & 11.3020 & 0 & 0 \\
\ldots & -1.5143 & -0.8669 & -3.5245 & 1.8519 & -2.8612 & -1.6440 & 10.3223 & 0
\end{array}\right]$$

Figure 3.10: Matrix $\mathbf{L}^{-1}$ where $\left(\mathbf{R}^t\mathbf{R}\right)^{-1} = \mathbf{L}\mathbf{L}^t$ is Cholesky factorization of $\left(\mathbf{R}^t\mathbf{R}\right)^{-1}$

With 125 initial values for the fmincon function, we could recover 11 rows of the secret basis. From the 30 rows of the result matrix, 18 of them, in pairs, correspond to ± of one row of the secret basis. Two of the rows correspond to a row of secret basis but are not in ± pairs. 10 rows that do not correspond to any vector of the basis, either completely differ from the vectors or have some different values (marked in boxes).

$$\left[\begin{array}{cccccccccccccccc}
\longrightarrow & -4 & 0 & 4 & -4 & 2 & 0 & 0 & -3 & 1 & 4 & -1 & -4 & -2 & 4 & 0 \\
 & 4 & 0 & -4 & 4 & -2 & 0 & 0 & 3 & -1 & -4 & 1 & 4 & 2 & -4 & 0 \\[1em]
\longrightarrow & 3 & 1 & -2 & -2 & 1 & -3 & 2 & 1 & -3 & 4 & -2 & -3 & 0 & -1 & -4 \\
 & -3 & -1 & 2 & 2 & -1 & 3 & -2 & -1 & 3 & -4 & 2 & 3 & 0 & 1 & 4 \\[1em]
\longrightarrow & -3 & 2 & 3 & 4 & -1 & 0 & -3 & -1 & 3 & -2 & 2 & 2 & 4 & 4 & -1 \\
 & 3 & -2 & -3 & -4 & 1 & 0 & 3 & 1 & -3 & 2 & -2 & -2 & -4 & -4 & 1 \\[1em]
\longrightarrow & -3 & 4 & -4 & 4 & -4 & 4 & 4 & -2 & -2 & -1 & 2 & 2 & -4 & 3 & 2 \\
 & 3 & -4 & 4 & -4 & 4 & -4 & -4 & 2 & 2 & 1 & -2 & -2 & 4 & -3 & -2 \\[1em]
\longrightarrow & -2 & -2 & 4 & 2 & -1 & 4 & 1 & 1 & -1 & -1 & 4 & 1 & 2 & -2 & 1 \\
 & 2 & 2 & -4 & -2 & 1 & -4 & -1 & -1 & 1 & 1 & -4 & -1 & -2 & 2 & -1 \\[1em]
\longrightarrow & -2 & -1 & -2 & -1 & -1 & 2 & -1 & -1 & 0 & 3 & -3 & 3 & 4 & -4 & -3 \\
 & 2 & 1 & 2 & 1 & 1 & -2 & 1 & 1 & 0 & -3 & 3 & -3 & -4 & 4 & 3 \\[1em]
\longrightarrow & -1 & -2 & -4 & 0 & 3 & -2 & -1 & 2 & -3 & 3 & 0 & -4 & -2 & 1 & -1 \\
 & 1 & 2 & 4 & 0 & -3 & 2 & 1 & -2 & 3 & -3 & 0 & 4 & 2 & -1 & 1 \\[1em]
\longrightarrow & 0 & -1 & 2 & -3 & 4 & 3 & -1 & -3 & 2 & -4 & -3 & -1 & -4 & 1 & -2 \\
 & 0 & 1 & -2 & 3 & -4 & -3 & 1 & 3 & -2 & 4 & 3 & 1 & 4 & -1 & 2 \\[1em]
\longrightarrow & 0 & 0 & -1 & 1 & 4 & -4 & -2 & -1 & 2 & 4 & -2 & -3 & 0 & -2 & 2 \\
 & 0 & 0 & 1 & -1 & -4 & 4 & 2 & 1 & -2 & -4 & 2 & 3 & 0 & 2 & -2 \\[1em]
\bullet & 1 & 4 & 3 & 4 & 4 & 2 & -2 & 0 & 0 & 3 & 1 & -2 & -3 & 2 & -4 \\[1em]
\bullet & 3 & 3 & -1 & 0 & 0 & -4 & 1 & 2 & -2 & 1 & 3 & -4 & 3 & 4 & -4 \\[1em]
 & 0 & 0 & 1 & \boxed{\mathbf{0}} & -4 & 4 & 2 & 1 & -2 & -4 & 2 & 3 & 0 & 2 & -2 \\
 & -3 & -1 & 2 & 2 & -1 & 3 & -2 & -1 & 3 & -4 & 2 & 3 & 0 & 1 & \boxed{\mathbf{3}} \\
 & -1 & -3 & -1 & 1 & 1 & 2 & 0 & 2 & -3 & 1 & 2 & -1 & 0 & -1 & 0 \\
 & -2 & -2 & -4 & 0 & 3 & -2 & -1 & 1 & -2 & 4 & 0 & -4 & -1 & 1 & -1 \\
 & -1 & 0 & -2 & -1 & 1 & 2 & 1 & -2 & 0 & -3 & -2 & -1 & -2 & 1 & -1 \\
 & -1 & 0 & -1 & 1 & 3 & -2 & -2 & -2 & 3 & 3 & -2 & -1 & -1 & -2 & 3 \\
 & 1 & 1 & -4 & 3 & -4 & -4 & 1 & 3 & -2 & 4 & 3 & 1 & 3 & -1 & 2 \\
 & 1 & 2 & 4 & -1 & -3 & 1 & 1 & -2 & 3 & -2 & -1 & 3 & 1 & 0 & 1 \\
 & 1 & 3 & -2 & 0 & -4 & 1 & 2 & -1 & 2 & 0 & 0 & 4 & 0 & -3 & 4 \\
 & 2 & 1 & 4 & -1 & 1 & 1 & -1 & 0 & 4 & 0 & 0 & 1 & 1 & -1 & 0
\end{array}\right]$$

Figure 3.11: Each row is equal to a $\mathbf{cL}^{-1}$; marked tuple rows are those that correspond to ± of a vector of the secret basis

CHAPTER 4

Conclusion

The need for cryptosystems that will resist quantum attacks is undeniable and lattice based cryptographic construction are promising candidates for such systems. Despite the fact that efficient cryptographic systems that have provable security have not been designed and standardized yet, there sure will be great improvements considering recent concentration on lattice based cryptography.

In this study, we first provided a survey on necessary background on lattices, lattice basis, computational problems and results of applying lattice reduction algorithms for solving these problems. Next we analyzed a signature scheme which has provable security and relies on hardness of approximating the shortest vector on ideal lattices. We then studied the GGH signature scheme, which is based on Closest Vector Problem and needs $O(n^2)$ time and space for its operation. NTRUSign was studied next; a signature scheme which like GGH signature is based on hardness of solving the Closest Vector problem but since it uses compact NTRU lattice, it needs less space and time by a factor of n compared to GGH signature scheme. Next we studied a successful cryptanalysis of GGH signature and NTRUSign, which exploits the fact that these two signature schemes lack zero knowledge property and with every message that they sign, they leak some information about the secret value. And finally, we implemented this attack on an 15 dimensional lattice and we could recover 11 vectors of the secret basis using MATLAB's fmincon function and 125 initial values and 5000 samples from the fundamental parallelepiped.

REFERENCES

[1] M. Ajtai, *Generating Hard Instances of Lattice Problems*, 28th ACM Symposium on Theory of Computing, Philadelphia, pp 99–108, 1996.

[2] L. Babai, *On Lovasz' lattice reduction and the nearest lattice point problem*, Combinatorica, Vol. 6, pp 1–13, 1986.

[3] D. J. Bernstein, *Introduction to post-quantum cryptography*, Post-Quantum Cryptography, 2009.

[4] C. Gentry, M. Szydlo,*Cryptanalysis of the revised NTRU signature scheme*, In Proceedings of Eurocrypt 2002, Lecture Notes in Computer Science, Vol. 2332, 2002.

[5] O. Goldreich, S. Goldwasser, S. Halevi. *Public-key cryptosystems from lattice reduction problems*, In Proceedings of Crypto '97, Lecture Notes in Computer Science. Vol. 1294 pp 112–131, 1997.

[6] J. Hoffstein, J. Pipher, J. H. Silverman, *NTRU: A ring-based public key cryptosystem*, Algorithmic Number Theory, Lecture Notes in Computer Science, Vol. 1423, pp 267-288, 1998.

[7] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, W. W. NTRUSign, *Digital Signatures Using the NTRU Lattice*, Topics in Cryptology, CT-RSA, Lecture Notes in Computer Science, Vol. 2612, pp 122-140, 2003.

[8] A. Korkine and G. Zolotareff, *Sur les forms quadratiques*, Math. Annafen, Vol. 6, pp 366-389, 1873.

[9] J. C. Lagarias, H. W. Lenstra Jr., C. P. Schnorr, *Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice*, Combinatorica, Volume 10, Issue 4, pp 333-348, 1990.

[10] A. K. Lenstra, H. W. Lenstra Jr., L. Lovasz, *Factoring polynomials with rational coefficients*, Mathematische Annalen, Vol. 261, Issue 4, pp 515-534, 1982.

[11] C. Ludwig, *A faster lattice reduction method using quantum search*, In Algorithms and Computation, Lecture Notes in Computer Science Vol. 2906, pp 19-208, 2003.

[12] V. Lyubashevsky, *Towards Practical Lattice-Based Cryptography*, Ph.D. Thesis, 2008.

[13] V. Lyubashevsky, D. Micciancio, *Generalized Compact Knapsacks Are Collision Resistant* Automata, Languages and Programming, Lecture Notes in Computer Science, Vol. 4052, pp 144-155, 2006.

[14] V. Lyubashevsky, D. Micciancio, *Asymptotically Efficient Lattice-Based Digital Signatures*, Theory of Cryptography, Lecture Notes in Computer Science, Vol. 4948, pp 37-54, 2008.

[15] J. Martinet, *Perfect Lattices in Euclidean Spaces*, Grundlehren der mathematischen Wissenschaften, Vol. 327, Springer, New York, 2003.

[16] D. Micciancio, *The shortest vector problem is NP-hard to approximate to within some constant*, SIAM Journal on Computing, Vol. 30, No. 6, pp 2008–2035, 2001.

[17] D. Micciancio, S. Goldwasser, *Complexity of Lattice Problems, A Cryptographic Perspective*, The Kluwer International Series in Engineering and Computer Science, Vol. 671, Kluwer Academic Publishers, Boston, Massachusetts, 2002.

[18] D. Micciancio, *Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions*, Computational Complexity, Vol. 16, No.4, pp 365-411, 2007.

[19] P. Q. Nguyen, O. Regev, *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures*, Advances in Cryptology, EUROCRYPT, Lecture Notes in Computer Science, Vol. 4004, pp 271-288, 2006.

[20] J. Proos, C. Zalka, *Shor's discrete logarithm quantum algorithm for elliptic curves*, Quantum Information and Computation, Vol. 3 No. 4, pp 317-344, 2003.

[21] O. Regev, *Lattices in computer science*, Lecture notes of a course given in Tel Aviv University, 2004.

[22] C. P. Schnorr, *A hierarchy of polynomial time lattice basis reduction algorithms*, Theoretical Computer Science,Vol. 53, No. 2-3, pp 201-224, 1987.

[23] P. Schnorr, *Block reduced lattice bases and successive minima*, Combinatorics, Probability and Computing, Vol. 3, pp 507–522, 1994.

[24] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, Vol. 26 No. 5, pp 1484–1509, 1997.

[25] M. Wiener, *Cryptanalysis of short RSA secret exponents*. IEEE Transactions on Information Theory, Vol. 36 No. 3, pp 553–558, 1990.