Project 1
Lattice-based Cryptography

Xiuqi Zhang Zikun Zhou Anna Li
Heyin Shen
UM-SJTU Joint Institute

Friday 25th June, 2021

Abstract

Lattice-based cryptography is a general term of cryptography that involves lattices. Lattice-based cryptosystem is post-quantum secure (still), and have better proved security, which is based on worst-case difficult lattice problems. It does not involve complex calculation, which make lattice-based cryptosystem efficient. This report discussed through development of lattice theory, computation problems based on lattice, cryptosystems and possible vulnerabilities.

1 Introduction

Lattice-based cryptography is a general set of cryptography that involves lattices. It covers all cryptography fields that applied the idea of lattices, including in implementation, validation or security proof. The most important key is to connect the difficulty of cracking such cryptography to the difficulty of certain hard lattice problems. Lattice-based cryptosystems covers encryption, signatures and hash functions.

What makes lattice unique and more important than other cryptography methods, is that Lattice-based cryptosystems are (still) post-quantum computing secure, and have proved security basing on worst-case scenario.

In the following sections, we will first introduce the maths involved in lattice, especially about the part connected to cryptography. Then we will discuss the difficulty of different lattice problems, as well as why they are important to cryptography. We will further discuss two important problems, SIS and LWE which is most widely used hard-problems as basis for lattice cryptosystems. Finally, we will discuss practical cryptosystems and do crypto-analysis to them.

2 Lattice

Lattice, in the manner of geometry and group theory, is a subgroup of additive group $\mathbb{R}^n$ which is isomorphic to additive group $\mathbb{Z}^n$, and can span to real vector space $\mathbb{R}^{n1}$. It can also be viewed as subgroup of the set of all linear combination of basis of $\mathbb{R}^n$ that has integer coefficients. To make it easier to understand, we following discuss it with examples in Euclidean spaces, i.e. 2 dimention with cartesian corrdinates.

Less formally (while not indicating less accurate), lattice can be viewed as a set of points

$$L = \{a_1v_1 + a_2v_2 + \cdots + a_nv_n | a_i \in \mathbb{Z}\} \tag{2.1}$$

where $(v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$ and they are linear independent. An example is as Figure 1 where all points forming the lattice can be generated by linear combination with integer coefficients. We denote the set $B = \{v_1, v_2, \ldots, v_n\}$ as basis of the latice. Then the lattice can also be denoted as $L(B)$. Apparently basis is not unique.

2.1 Equivalent Bases

The very immediate problem raiseed will be how do we know whether bases generates the same lattice.

2.1.1 Column view

- Changing order of $\forall v_i, v_j \in B$ does not chagne the lattice generated.

[1]For convinience, we emit zero vector in following discussion if not specified

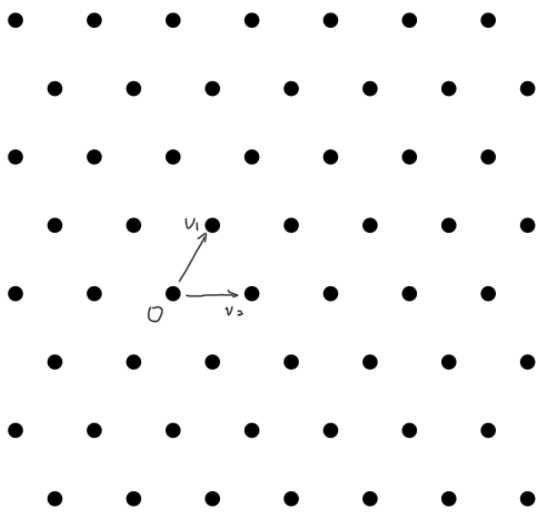

Figure 1: Lattice Example

- $\forall v_i \in B, L(B') = L(B)$ where $B' = (B/v_i) \cup \{-v_i\}$.
- Linear Combination: for some $v_i, v_j \in B$, let $v_i = v_i + kv_j$ where $k \in \mathbb{Z}$.

2.1.2 Matrix view

Theorem 2.1.

$$L(B_1) = L(B_2) \iff B_1 = B_2U \tag{2.2}$$

where U is a unimodular U.

Proof. **If** $\mathcal{B}_1 = \mathcal{B}_2$, we know that for all columns $b_{1_i} \in B_1$, there exists some matrix U_2 such that $B_1 = B_2U$. Also we can know that there exist U_1 such that $B_2 = B_1U_1$. Thus,

$$B_2 = B_1U_1U_2 \tag{2.3}$$

Consider following equation:

$$B_2^TB_2 = (U_1U_2)^TB_2^TB_2(U_1U_2) \tag{2.4}$$

it's determinants equation is

$$\det(B_2^TB_2) = (det(U_1U_2))^2 \det(B_2^TB_2) \tag{2.5}$$

which idicates that $\det(U_1U_2) = \pm 1$. Since $\det(U_1)\det(U_2) \in \mathbb{Z}$, we know that $\det(U_1)\det(U_2) \in \{-1, 1\}$

If $B_2 = B_1U$ where U is a unimodular matrix. From defination of lattice we know that $\mathcal{L}(B_2) \subseteq \mathcal{L}(B_1)$. Consdier that $B_1 = B_2U^{-1}$, which means that reverse also holds. Thus, $\mathcal{L}(B_1) = \mathcal{L}(B_2)$. □

2.2 Lattice meaning to space

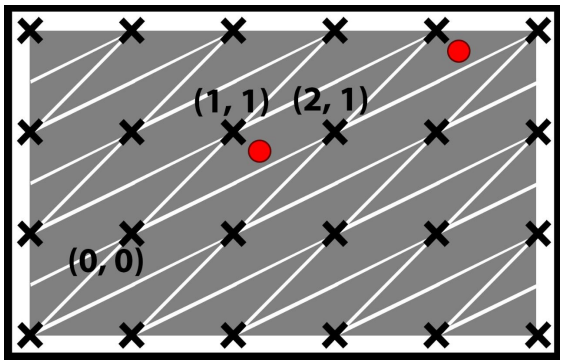

Figure 2: Dividing space

Lattice can be used to denote equal repeating division to a space, and a special case is the basic division in shape of parallelepiped (maybe more than 3-dimention) starting from origin point can be expressed as

$$\mathcal{B} = \{a_1b_1 + a_2b_2 + \cdots + a_nb_n | a_i \in [0, 1)\} \tag{2.6}$$

Notice that all these divisions are identical, so from such lattice all points in space can be mirrored in to the fundamental block, and also get its relative location in its division

$$\begin{aligned} p &= a_1b_1 + a_2b_2 + \ldots a_nb_n \\ &\to p \mod P(B) \\ &= (a_1 \mod 1)b_1 + (a_2 \mod 1)b_2 + \cdots + (a_n \mod 1)b_n \end{aligned} \tag{2.7}$$

in which way we can use lattice to show repeated patterns across space.

Notice that no matter which basis is chosen, the fundamental parallelpiped has the same volume. This can be proved by imagining a very large space where the

shape of each small region can be ignored. Since the number of small regions is the same the volumn of them should be the same.

We define the determinant of a lattice $L(B)$ as $\det(L) = |\det(B)|$, which is the volume of the fundomental parallelepiped.

2.3 Successive Minima

One very important element of a lattice is the shortest vector in the lattice. We denote the length (Euclidean norm) of the shortest vectors in $\mathcal{L}$ as $\lambda_1(\mathcal{L})$, the second shortests as $\lambda_2(\mathcal{L})$,..., etc.

2.4 Gram-Schmidt Orthogonalization

Gram-Schmidt Orthogonalization is a process which takes a set of linearly independent vectors and output a set of orthogonal vectors with same cardinality. It projects each vector on the orthogonal complement of the previous vectors. A formal design is as Eq. 2.8

For vector series $B = b_1, b_2, \ldots, b_n$, GSO vector set $\widetilde{B} = \tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_n$ is as

$$\tilde{b}_i = b_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{b}_j, \text{ where } \mu_{i,j} = \frac{\left\langle b_i, \tilde{b}_j \right\rangle}{\left\langle \tilde{b}_j, \tilde{b}_j \right\rangle} \tag{2.8}$$

Notice that $\text{span}(B) = \text{span}(\tilde{B})$ while $\tilde{B}$ is not necessarily basis of $\mathcal{L}$.

By normalizaing $\tilde{B}$, we can get an basis which is orthonormal, and can be shown as matrix as

$$\begin{pmatrix} \left\|\tilde{b}_1\right\| & k_{2,1}\left\|\tilde{b}_1\right\| & \cdots & k_{n,1}\left\|\tilde{b}_1\right\| \\ 0 & \left\|\tilde{b}_2\right\| & \ldots & k_{n,2}\left\|\tilde{b}_n\right\| \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 0 & \left\|\tilde{b}_n\right\| \\ 0 & \ldots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \end{pmatrix} \tag{2.9}$$

By observing this matrix with lots of zero, we can conclude that the volume of $\mathcal{P}(b_1, \ldots, b_n)$ is as

$$\det(\mathcal{L}(b_1, \ldots, b_n)) = \prod_{i=1}^{n} \|\tilde{b}_i\| \tag{2.10}$$

Theorem 2.2. $\lambda_1 \geq \min\|\tilde{b}_i\|$

Proof. Notice that this is a upper-right matrix, thus there must exist norm of one column that can't be eliminated when calculating determinant. □

2.5 Minkowski's Theorem

Lemma 2.3. *For any lattice* Λ *and set* S *of region with volume larger than* $\det(\Lambda)$*, than* $\exists z_1, z_2 \in S, z_1 \neq z_2 \in \Lambda$.

Proof. Imagine a significantly large space, where size of regions can be ignored. Then set S must take more space than same amount of fundamental parallelepiped space if there is no overlap. □

Theorem 2.4. ***Minkowski's Theorem:*** *For any lattice* Λ *and convex zero-symmetric set* S*, volume of which is larger than* $2^n \det(\Lambda)$*, there must exists some lattice point in* S*. (which is the upper bound of smallest lattice).*

Proof. We first consider two point which is in set S', which is generated by shrinking all dimention of S by two. Denote them as s_1 and s_2.

Since S is a convex set, $2s_1$ and $-2s_2$ should both be in S, which means the mid point of them, which is $s_1 - s_2$, is in S. □

Theorem 2.5. ***Inference of Minkowski's Theorem:***

$$\forall \Lambda, \lambda_1(\Lambda) \leq \sqrt{n} \cdot \det(\Lambda)^{\frac{1}{n}} \tag{2.11}$$

Proof. It is merely adding the factor of dimention to make both sides factor the same size from Theorem 2.4 while only considering the special case of ball. □

3 Basic Computation Lattice Problems

3.1 Shortest Vector Problem (SVP)

Shortest[2] Vector Problem is the most important and basic computation problem about lattice. From Theorem 2.4 and Theroem 2.2 we know about the upper and lower bound of shortest vector, however it does not provide a way to find such vector. This problems remains to be a hard problem, and works in the field of lattice compuatation problems as basic as SAT problem in NP-complete.

Since this is a very difficult problem (which will be shown afterwards), an approximate problem with factor γ is worth considering, where instead of finding $\lambda(\mathcal{L}(B))$, we try to find some vector v which $\|v\| \leq \gamma \cdot \lambda(\mathcal{L}(B))$. This easier problem is called as SVP_γ.

3.1.1 Hardness

In Euclidean distance, which as most scenario this report is discussed on, we only know that by applying randomized reductions the problem is NP-hard [2]. If considering uniform norm, the problem has already been proved to be NP-hard [1].

3.2 GapSVP

GapSVP_γ is variant of SVP_γ, in which we try to know that whether $\lambda(\mathcal{L}(B))$ is not bigger than one, or larger than γ, where γ is some function $f(n)$, where n is the dimension of the space. Notice that it is a promise problem, which means input should make sure the result will in one of the conditions.

3.3 Closest vector problem (CVP)

The closest vector problem is about in following scenario:

A basis B and a lattice L, and some vector $v \in \text{span}(B)$.

Try to find:

[2]If not specified, all discussion about length in this report is Euclidean norm.

A vector $v' \in L$, which is closest to v.

Similarly we have CVP_γ where we try to find the vector with distance smaller than $\gamma\cdot$ smallest distance.

First thing to notice about CVP is that, given an algorithm of CVP, we can solve SVP in polynomial time which is a very small polynomial factor. This can be done by trying to find closest vector to very small vectors.

3.3.1 Hardness

Further from the conclusion that we can solve SVP efficiently if we can solve CVP, Goldreich et al. proved that CVP is at least harder than SVP at any aspect [3]. Dinur et al. proved that, with factor $n^{c/\log\log n}$ for some constant $c > 0$, CVP is NP-hard to approximate [5].

3.4 Summary

In summary, the hardness about the problem (currently) according to factor is as Figure 3.

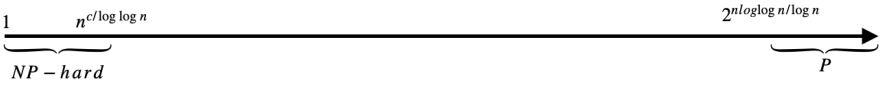

Figure 3: Hardness according to factor

4 Advantage of Lattice-based cryptography

As the an important class in the public-key post-quantum cryptography, Lattice-based cryptography have the leading advantages in the following aspects.

Anti-quantum attack This is the main advantage the lattice-based cryptography has over the traditional public-key cryptographies as the security of the latter has been challenged in the context of a quantum computer. In fact, the security guarantee of most traditional public-key cryptographies is established based

on the hardness of the factorization of large integers, the discrete logarithm and other related problems. However, with the proposition of the quantum algorithm, both factoring and discrete algorithm-based problems are solvable in polynomial time complexity. Therefore, in the research of modern public-key cryptosystem, lattice-based cryptography outstands for its ability to resist quantum attacks.

4.1 Efficient algorithm and high concurrency

The main mains problems involved in the lattice-based cryptosystem are based on the calculation of vectors without the engagement of large prime integers, and the algorithm also enjoys relative high concurrency, leading to high efficiency in practice.

4.2 Worst case to average case reduction

The security of the lattice-based cryptography can be guaranteed because it is built based on the "worst case to average case reduction". In other words, the hardness of finding the solutions of a certain problems in average cases is no less than finding the solutions in the worst cases. In practice, efficient lattice-based algorithms, such as those that are based on LWE, their worst-case hardness results may be unknown. Conversely, cryptographic that are based on factoring, which we know is hard in the worst case, can still be decrypted easily when it is easy to solve the factorization on average input. While as lattice-based cryptosystem is hard to solve in the worst case, it possesses very high security.

However, there are still some disadvantages for lattice-based cryptography and the main problem to conquer is that the existing lattice-based cryptosystem suffers from an unsatisfactorily large length of the private key. Though this situation has progressed, there still remains room for further improvement. Following we are going to discuss two kinds of Lattice problem, SIS and LWE, which is most commonly used in cryptography.

5 Shortest Integer Solution Problem (SIS)

The short integer solution problem (SIS) is an average-case problem based on the worst-case lattice problem. Its difficulty is guaranteed by the short vector problem

(SVP) from Section 3.1. It is proposed by Ajtai in 1996 in order to transform a geometric lattice problem into an integer problem so that it can be processed by computer program.

5.1 Definition of the SIS

Given m random vectors $a_1, a_2, ..., a_m$ in $\mathbb{Z}_q^n$ (e.g., $q \approx n^3$), find a non-trivial solution $z_1, z_2, ..., z_m$ in $\{-1, 0, 1\}$ such that:

$$z_1 \cdot \begin{pmatrix} | \\ a_1 \\ | \end{pmatrix} + z_2 \cdot \begin{pmatrix} | \\ a_2 \\ | \end{pmatrix} + \cdots + z_m \cdot \begin{pmatrix} | \\ a_m \\ | \end{pmatrix} = \begin{pmatrix} | \\ 0 \\ | \end{pmatrix} \in \mathbb{Z}_q^n \tag{5.1}$$

We can rewrite this equation in a matrix form, Denote $(a_1, a_2, ..., a_m)$ as $\mathbf{A}$ in $\mathbb{Z}_q^{n \times m}$ and the solution $z_1, z_2, ..., z_m$ as $\mathbf{Z}$ in $\mathbb{Z}^m$. Then the SIS problem is to find a short vector $\mathbf{z} \in \{-1, 0, 1\}^m$ such that:

$$\begin{pmatrix} \cdots & \mathbf{A} & \cdots \end{pmatrix} \begin{pmatrix} \\ \mathbf{z} \\ \\ \end{pmatrix} = 0 \in \mathbb{Z}_q^n \tag{5.2}$$

There is a relationship between the SIS and the lattice problems:
Let S be the set of all solution $\mathbf{z}$, such that $\mathbf{Az} = 0 \pmod q$. As this set is additive, S is a lattice. Therefore, the SIS problem can be regarded as the problem to find a short vector in S.
Now we have a new representation of lattices constructed from matrix $\mathbf{A}$:

$$L^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = 0 \pmod q\} \tag{5.3}$$

Using worst-case to average-case reduction, solving the SIS problem in $L^{\perp}(\mathbf{A})$ is approximately solving SVP in all lattices, which will be discussed later.

5.2 One-Way & Collision-Resistant Hash Function using SIS

An one-way & collision-resistant hash function can be easily implied from the SIS: Set $m > n \lg q$. Given random $\mathbf{A}$ in $\mathbb{Z}_q^{n \times m}$, define the hash function $f_{\mathbf{A}} : \{0, 1\}^m \to \mathbb{Z}_q^n$ as:

$$f_{\mathbf{A}}(\mathbf{z}) = \mathbf{Az} \tag{5.4}$$

A collision $f_{\mathbf{A}}(\mathbf{z}) = f_{\mathbf{A}}(\mathbf{y})$ yields a solution $\mathbf{z} - \mathbf{y}$ of SIS for $\mathbf{A}$, as $\mathbf{z} - \mathbf{y}$ is in $\{0, 1\}^m$ and satisfy $\mathbf{A}(\mathbf{z} - \mathbf{y}) = 0$

5.3 Worst-case to Average-case Reduction

5.3.1 Uniform Distribution Over Lattices

Lemma 5.1. *Consider a Gaussian distribution:*

$$\rho_s(x) = (1/s)e^{-\pi x^2/s^2} \tag{5.5}$$

and s=5M, for some positive M, if $X \sim \rho_s$*, then for all* $m < M$*:*

$$\Delta(X \mod m, \text{Uniform}[0, m)) < 2^{-110} \tag{5.6}$$

Using this lemma, we can generate uniform elements on a line segment using Gaussian distribution. In order to generate uniform random vectors in the Lattice space to generate $\mathbf{A}$, we need to extend this lemma to the multidimensional space. Then we have:

Theorem 5.2. *If* $s > 5\lambda_n(\mathbf{B})$*, and* $\mathbf{X} \sim \rho_s(\mathbf{x}) = (1/s)^n e^{-\pi||\mathbf{x}||^2/s^2}$*, then*

$$\Delta(\mathbf{X} \mod \mathbf{B}, \text{Uniform}(\mathbf{B})) < n2^{-110} \tag{5.7}$$

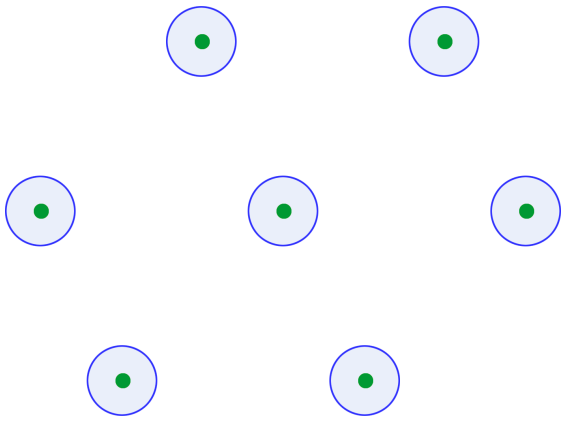

Figure 4: The distribution when s is small

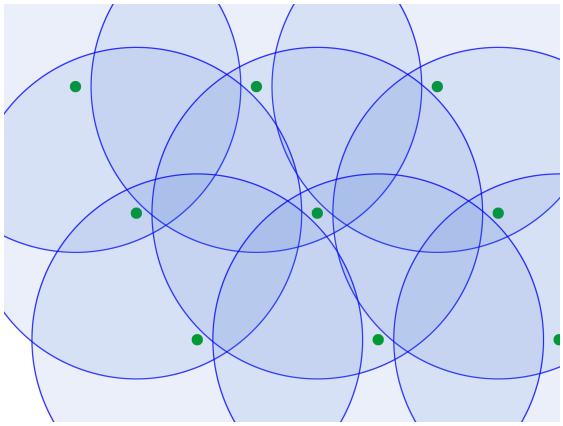

Figure 5: The distribution when s increases

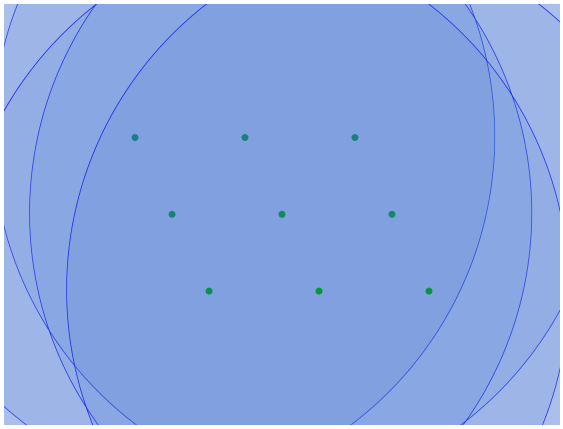

Figure 6: The distribution when s is large enough

These pictures demonstrate the Theorem 5.2. As the deviation s increases, the distribution becomes uniform over lattices.

5.3.2 The Reduction

Now considering the hardest SVP with lattice $L(\mathbf{B})$. It can be solved using an SIS oracle:

Algorithm 1 Solving SVP using SIS oracle

for m times **do**
  Pick a random lattice point v_i
  Gaussian sample a point $a_i = v_i + r_i$ round to $\mathbb{Z}_q^n$ around v_i
end for
$\mathbf{A} = (a_1, a_2, ..., a_m) \rightarrow$ SIS oracle
SIS oracle $\rightarrow \mathbf{z}$
Output the short lattice vector: $\mathbf{Rz}$

1. Theorem 5.2 indicates that $\mathbf{A} = (a_1, a_2, ..., a_m)$ is uniformly random in $\mathbb{Z}_q^n$, Therefore, we can give $\mathbf{A}$ to the SIS oracle.

2. The SIS oracle will output the solution of $\mathbf{Az} = 0$. Let $\mathbf{V} = v_1, ..., v_m$ and $\mathbf{R} = r_1, ..., r_m$, $\mathbf{Az} = 0$ is a zero vector which is a lattice vector in $L(\mathbf{B})$ and $\mathbf{Vz}$ is a lattice vector. Therefore, $\mathbf{Rz}$ is also a lattice vector.

3. As $\mathbf{z} \in \{-1, 0, 1\}^m$ and r_i is short, $\mathbf{Rz}$ is a short lattice vector which is the solution of the SVP.

Therefore, the SIS problem is hard, provided the SVP is hard to approximate for some hardest $L(\mathbf{B})$.

6 Learning With Errors Problem (LWE)

LWE is a computation problem used for most Lattice-based cryptosystems. The problem is about trying to find a linear n-ary function f which is over a finite ring from some possible to be error sample pair $y_i = f(\mathbf{x}_i)$. A formal design is given below.

6.1 Definition of LWE

Consider an additive group $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ which is constructed modulo one. For error, produce a fixed probability distribution over $\mathbb{T}$ denoted as φ.

We than define a distribution over $\mathbb{Z}_q^n \times \mathbb{T}$ as

1. Randomly get a vector $a \in \mathbb{Z}_q^n$ following uniform distribution.

2. Randomly get a number $\epsilon \in \mathbb{T}$ following distrubtion φ.
3. Compute addtion and division under $\mathbb{T}$, inner product in $\mathbb{Z}_q^n$ calculate $t =< a, s > /q + e$.
4. Pair (a, t) is a sample.

Denote the entire sample set as $A_{s,\varphi}$.

6.1.1 Search problem

With above definition, we define the LWE search problem as trying to find s, given polynomial amout of samples from $A_{s,\varphi}$. Most times we studied a special case of LWE, where φ is the nomal distrubtion as origin point with variance of $\frac{\alpha^2}{2\pi}$, i.e. $e^{-\pi(|x|/\alpha)^2}/\alpha$.

6.1.2 Decision problem

On the other hand, LWE decision problem is to tell the difference between a LWE distributed input and a uniformly random input.

6.1.3 Equivalent proof of Search and Decision problem of LWE

Regev showed that they are equivalent, we will breifly discuess the proof following [10].

From Search to decision If we assume that we have an algorithm that can solve search problem, we solve decision problem by observing

$$C = < a_i, s > -t_i \tag{6.1}$$

It will have identical distribution with sample, i.e. if it is in uniform distribution the sample is in distrubtion and vice versa.

Algorithm 2 LWE decision to search

```
for each coordinate of s, s_1 do
  while result is not correct do
    Guess k ∈ Z_q
    Pick a random number r ∈ Z_q.
    for each sample pair (a_i, b_i) do
      a_i ← a_i + (r, 0, 0, ..., 0)
      b_i ← b_i + rk/q
    end for
    result = Alg(a_i, b_i)
  end while
end for
```

From Decision to Search If we have an algorithm Alg: $\mathbb{Z}_q^n \times T \to \{True, False\}$ for decision problem, we can solve search problem as Algo 2.

Since q is prime, it is bounded as polynomial of n, which means we can get approximate guess for every s_i in polynomial time.

6.2 Average Hardness

Peikert proved that the worst case of LWE can be reducted to GapSVP in polynomial time, considering a approximate output [8].

7 Ring learning with errors key exchange

The ring learning with errors key exchange (RLWE-KEX) is a typical example of the lattice-based cryptosystem, which has the speciality to be reduced to a known hard problem. The process is carried out in the environment of the ring of polynomials modulo a polynomial $\Phi(x)$ and the coefficients of the polynomials are in the field of integers modulo a prime q. The $\Phi(x)$ is the nth cyclotomic polynomial defined as $\Phi_n(x) = \prod_{\substack{1 \le k \le n \\ \gcd(k,n)=1}} \left(x - e^{2i\pi\frac{k}{n}}\right)$. One important factor to maintain its security is whether the system can generate random polynomials. Usually, with the help of uniform sampling and discrete Gaussian sampling, we are able to generate random coefficients and thus generating the random polynomials. Two devices are involved in the key exchange process, among which one is called an initiator (I) and another is called a respondent (R). Both of them are aware

of the values of q, n, $a(x)$ where $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$, and they can generate polynomials based on the discrete Gaussian distribution χ_α on the ring $R_q = Zq[x]/\Phi(x)$. The detailed function of this algorithm is composed of two main parts: the Initiation as Algo 3 and the Response as Algo 4.

Algorithm 3 Initiation

s_I, $e_I \leftarrow polynomials\ with\ coefficients\ from\ \chi_\alpha\ distribution$
$p_I \leftarrow as_I + 2e_I$
return p_I

Algorithm 4 Response

$\mathbf{E} \leftarrow \left\{-\left\lfloor \frac{q}{4} \right\rfloor, \ldots, \left\lfloor \frac{q}{4} \right\rceil\right\} of Zq = \left\{-\frac{q-1}{2}, \ldots, \frac{q-1}{2}\right\}$
$s_R, e_R \leftarrow polynomials\ with\ coefficients\ from\ \chi_\alpha\ distribution$
$p_R \leftarrow as_R + 2e_R$
$e'_R \leftarrow sample from \chi_\alpha\ distribution$
$k_R \leftarrow p_I s_R + 2e'_R$
for each coefficient k_{R_i} of k_R **do**
  if $k_{R_i} \in E$ **then**
    $w_i \leftarrow 0$
  else
    $w_i \leftarrow 1$
  end if
end for
$sk_R = \left(k_R + w \cdot \frac{q-1}{2}\right) \bmod q \bmod 2$
return p_R, w

Then after the Initiator receives the p_R and w from the Responder, he/she can sample e'_I from χ_α and calculate $k_I = p_R s_I + 2e'_I = as_I s_R + 2e_R s_I + 2e'_I$, and finally get the key stream $sk_I = \mathrm{Mod}_2(k_I, w)$.
Some choices of parameters have been suggested that: n = 512, q = 25601, and $\Phi(x) = x^{512} + 1$ for 128 bits of security, and n = 1024, q = 40961, and $\Phi(x) = x^{1024} + 1$ for 256 bits of security. While improvements have been made to choose n = 1024, q =12289, and $\Phi(x) = x^{1024} + 1$, reducing 70% of the previous length of the public key.

8 Possible Attack for Lattice-based Cryptography

8.1 Fault Attacks

Although lattice-based Cryptography is the most safe algorithm, there still exists the implementation level fault, which makes the attack possible. Here we mainly talked about three fault attacks

8.1.1 Loop-Abort Faults on Lattice-based Signature

in 2018, Espitau proposed a loop-abort faults applied on lattice-based signature.[4] In this research, they applied two attack but with roughly the same type of faults, so that the attacker could lead to a loop inside the algorithm of the signature and abort early. The first attack is in the Fiat-Shamir family. By inputing a fault in the loop, they could get the commitment value, which is a random polynomial. This could leak enough information for recovering the entire signing key. For the GPV-based hash-and-sign signature scheme, when it is applied into the early loop abort, the original ciphertext will become a linear combination of the parts of the secret lattice. Therefore, in this way, we could recover the key by repeating this process.

8.1.2 "Fiat-Shamir with Aborts" Framework

Based on loop-abort faults, in 2019, Prasanna Ravi et etc. proposed "a practical fault attack on pqm4 Implementations of NIST candidates"[9].By using a "skip-addition"attack, the attaker could retrieve the primary secret. then the attack could make a forgery attack on the Dilithium signature scheme.
Supposing that the attacker could access to the device physically and could trigger the device many number of times. And we assume that target creates P signatures of $(z[i][j], c[i]), i \in \{1, ..., P\}, j \in \{0, ..., l-1\}$ Then the signature component z is

$$z[i][j] = s_1[j] \times c[i] + y[i][j]$$

And there are two types for the fault attacks
The first one is:

$$z = s_1 \cdot c \Rightarrow z = z + y$$

then after computing the t^{th} coefficient of z

$$\Rightarrow (\hat{z})_t =< s_1, Rot(c,t) >$$

The second one is:

$$z = y \Rightarrow z = z + s_1 \cdot c$$

then after computing the t^{th} coefficient of $s_1 c$is:

$$(s_1 c)_t =< s_1, Rot(c,t) >= (z)_t - (\hat{z})_t$$

After repeating this process for multiple runs, we could extract enough information to recover the primary secret.

8.2 Effective Attacks

8.2.1 Physical Attack

One classical way of attacking the lattice-based schemes is the physical attack, since there are little research results on the physical security of lattice-based cryptography. Moreover, physical attack is easier comparing to other attack. [4]

8.2.2 Cryptanalysis of GGH

In 2018,Yupu Hu and Huiwen Jia presented several efficient attacks on GGH.[6] In the attack testing experiments, the most important part is the specific modular operations so as they could reduce the effect of noise of GGH. In this way, with little lattice-reduction tools, MKE(multipartite key exchange) would be attacked. To break WE(witness encryption), they made use of the hardness of exact-3-call(X3C) problem. By combining these two steps, enough useful information for attack could be gotten and the attack towards GGH succeeds.

8.2.3 Attack against Cai-Cusick

Cai-Cusick is a lattice-based public-key cryptosystem, and one of the main character of it is that it has little data expansion. And in 2019, Yanbin Pan and Yingpu Deng proposed a way of ciphertext-only attack towards the Cai-Cusick crypto system. [7]

Algorithm 5 The Ciphertext-Only Attack

Input: $v_{\sigma(0)}, v_{\sigma(1)}, ..., v_{\sigma(m)}$: The public key
b: the maximum of the public key
C: the ciphertext
Output: $M = (a_0, a_1, ..., a_m)$: The corresponding message
"Failure": message corresponding to errors
Use public keys to compute the Gram-Schmidt orthogonalization vectors:$v^*_{\sigma(0)}, v^*_{\sigma(1)}, ..., v^*_{\sigma(m)}$
if $min_{0 \le i \le m} \parallel v^*_{\sigma(i)} \le b \rightarrow$"Failure" **then**
halt $3 - 8$
end if
$i := m$
repeat
$a_i := \left\lceil \frac{<v^*_{\sigma_{(i)}}, C>}{\|v^*_{\sigma_{(i)}}\|^2} \right\rceil$
$C := C - a_i v_{\sigma_{(i)}}, i := i - 1$
until $i < 0$
return $(a_0, a_1, ..., a_m)$

This algorithm has the computational complexity of $O(m^2 n)$, and according to the experiments, when $(m < 500, n < 300)$, this algorithm could achieve 100% probability of success.

References

[1] M. Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 99–108. ISBN: 0897917855. DOI: 10.1145/237814.237838. URL: https://doi.org/10.1145/237814.237838.

[2] Miklós Ajtai. "The Shortest Vector Problem in L2 is NP-Hard for Randomized Reductions (Extended Abstract)". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 10–19. ISBN: 0897919629. DOI: 10.1145/276698.276705. URL: https://doi.org/10.1145/276698.276705.

[3] I. Dinur et al. "Approximating CVP to WithinAlmost-Polynomial Factors is NP-Hard". In: *Combinatorica* 23.2 (2003), pp. 205–243. ISSN: 1439-6912. DOI: 10.1007/s00493-003-0019-y. URL: https://doi.org/10.1007/s00493-003-0019-y.

[4] Thomas Espitau et al. "Loop-Abort Faults on Lattice-Based Signature Schemes and Key Exchange Protocols." In: *IEEE Transactions on Computers* 67.11 (2018), pp. 1535–1549. DOI: 10.1109/TC.2018.2833119.

[5] Oded Goldreich et al. "Approximating shortest lattice vectors is not harder than approximating closest lattice vectors". In: *Information Processing Letters* 71.2 (1999), pp. 55–61. DOI: 10.1016/S0020-0190(99)00083-6.

[6] Yupu Hu and Huiwen Jia. "Cryptanalysis of GGH Map". In: *IEEE Transactions on Computers* 67.11 (2018), pp. 1535–1549. DOI: 10.1109/TC.2018.2833119.

[7] Yanbin Pan and Yingpu Deng. "A Ciphertext-Only Attack Against the Cai-Cusick Lattice-Based Public-Key Cryptosystem". In: *IEEE TRANSACTIONS ON INFORMATION THEORY* 57.3 (2011), pp. 1780–1785. DOI: 10.1109/TIT.2010.2103790.

[8] Chris Peikert. "Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract". In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 333–342. ISBN: 9781605585062. DOI: 10.1145/1536414.1536461. URL: https://doi.org/10.1145/1536414.1536461.

[9] Prasanna Ravi et al. "Exploiting Determinism in Lattice-based Signatures: Practical Fault Attacks on Pqm4 Implementations of NIST Candidates." In: *Proceedings of the 2019 ACM Asia Conference on computer and communications security*. AsiaCCS '19. Auckland, New Zealand: ACM, 2019, pp. 427–440. ISBN: 1450367526. DOI: 10.1145/3321705.3329821. URL: https://doi.org/10.1145/3321705.3329821.

[10] Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *J. ACM* 56.6 (Sept. 2009). ISSN: 0004-5411. DOI: 10.1145/1568318.1568324. URL: https://doi.org/10.1145/1568318.1568324.