VE475 Introduction to Cryptography

Lattice-based Cryptography

Jiang, Sifan 515370910040

Wu, Hao 515370910041

Su, Jingyu 516370910123

Song, Gaopeng 516370910227

June 19, 2019

1 Introduction

A quantum computer is a computer whose operation exploits certain very special transformations of its internal state based on the laws of quantum mechanics and under very carefully controlled conditions [?]. In theory, any particles, like atom, electron, photon, can be used for quantum computing. The reason why quantum computers can have higher computing performance than any regular computers is that one particle can be viewed as a binary 0, a binary 1, or a 0 and 1 at the same time because of "Quantum Superposition". In such way, the computer can conduct parallel computing very efficiently and try all the possible solutions in very short time to realize complex computation. Based on such theory, *Shor* discovered the quantum factoring algorithm with time complexity $O((\log N)^3)$, making factoring-based cryptosystems no longer secure.

New cryptosystems are needed especially nowadays, companies like IBM has developing their own quantum computers, such as *IBM Q System One*. One of the possible post-quantum cryptosystem solution is lattice-based cryptography.

Lattices have been introduced to the field of mathematics first in the 18^{th} century in number theory by mathematicians such as Gauss and Lagrange. The study of lattices was advanced by Minkowski in his "Geometry of Numbers".

In 1982, Arjen Lenstra, Hendrik Lenstra, and László Lovász introduced their famous "LLL" basis-reduction algorithm in *Factoring Polynomials with Rational Coefficients* [?], which is used for factoring integer polynomials.

In 1996, Miklós Ajtai issued *Generating hard instances of lattice problems* and introduced "worst-case to average-case reduction" for lattice problems , providing a cryptographic one-way function based on worst-case hardness conjectures [?]. And the Short Integer Solution problem (SIS) serve as a foundation of numerous lattice-based cryptographic protocols: For positive integer parameters n, m, and q, find a short non-zero solution $\mathbf{z} \in \mathbb{Z}^m$ to the honogeneous linear system $\mathbf{A}\mathbf{z} = 0 \mod q$ for uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ [?]. Another main problem, as the foundation, is the Learning With Errors problem.

1.1 General Lattices

A lattice is a set of points in n-dimensional space with a periodic structure , such as the one illustrated in Figure 1, which is a simple example shows the two-dimensional space lattice and two groups of possible, as long as vectors are independent, bases (black and grey ones) [?].

Formally, given n-independent vectors $\mathbf{b}_1, \cdots, \mathbf{b}_n \in \mathbb{R}^n$, the lattice generated by these vectors is the set of vectors

$$\mathcal{L}(\mathbf{b}_1, \cdots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b_i} \ : \ x_i \in \mathbb{Z} \right\}. \tag{1}$$

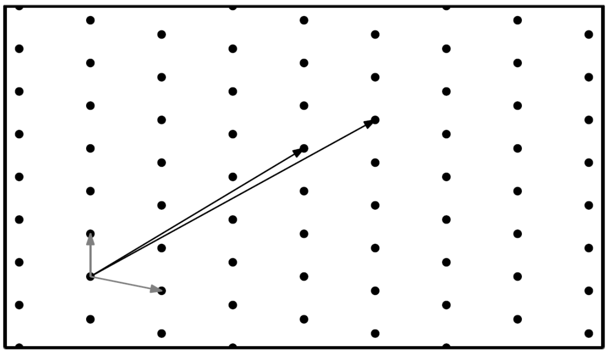

Figure 1: A two-dimensional lattice and two possible bases.

The vectors $\mathbf{b}_1, \cdots, \mathbf{b}_n$ are known as a basis, denoted $\mathbf{B}$, of the lattice. For example, $[1,5,-9]^T$, $[-2,2,0]^T$, and $[13,1,4]^T$ form an basis for $\mathbb{Z}^3$ and

$$\mathbf{B} = \begin{bmatrix} 1 & -2 & 13 \\ 5 & 2 & 1 \\ -9 & 0 & 4 \end{bmatrix}.$$

Notice that there exists multiple lattice bases which makes lattice-based cryptography possible. Any lattice can be obtained by applying some non-singular linear transformation to the integer lattice. Also, given $\mathbf{B}_1$, $\mathbf{B}_2$ two bases for lattice $\mathcal{L}$, there exist uni-modular matrices $\mathbf{U}$ such that $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}^{-1}$ [**?**].

Since lattice is in a periodic structure, the concept of fundamental region is used to formalize this idea. A set $\mathcal{F} \subseteq \mathbb{R}^n$ is a fundamental region of a lattice $\mathcal{L}$ if its translates $\mathbf{X} + \mathcal{F} = \{\mathbf{x} + \mathbf{y} \;:\; y \in \mathcal{F}\}$, taken over all $x \in \mathcal{L}$, form a partition of $\mathbb{R}^n$. And the fundamental parallelepiped of a lattice basis $\mathbf{B}$ is defined as

$$\mathcal{P}(\mathbf{B}) := \mathbf{B} \cdot \left[-\frac{1}{2}, \frac{1}{2}\right)^n = \left\{\sum_{i=1}^{n} c_i \mathbf{b}_i \;:\; c_i \in \left[-\frac{1}{2}, \frac{1}{2}\right)\right\}. \tag{2}$$

Then, the determinant of a lattice $\mathcal{L}$, denoted $\det(\mathcal{L})$, can be defined as $\mathrm{vol}(\mathcal{F})$.

Since a lattice $\mathcal{L}$ is discrete, it has two non-zero elements $\pm\mathbf{v} \in \mathcal{L}$ of minimum Euclidean distance. The exact definition of the minimum distance of a lattice $\mathcal{L}$ is defined as

$$\lambda_1(\mathcal{L}) := \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\|. \tag{3}$$

With Minkowski's First Theorem, we have for any lattice $\mathcal{L}$, we have $\lambda_1(\mathcal{L}) \leq \sqrt{n} \cdot \det(\mathcal{L})^{1/n}$. And using the exact formula for the volume of an n-dimensional ball, we can obtain a slightly tiger bound $\lambda_1(\mathcal{L}) \leq \sqrt{n/(2\pi e)} \cdot \det(\mathcal{L})^{1/n}$.

1.2 Lattice Problems

1.2.1 Shortest Vector Problem

Shortest Vector Problem (SVP) is the most important lattice-based computational problem, which requires the approximate of the minimal Euclidean length of a non-zero lattice vector. The definition of Approximated Shortest Vector Problem (SVP$_\gamma$) is: find a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}$ such that

$$\|\mathbf{v}\| \leq \gamma \cdot \min_{\mathbf{w} \in \mathcal{L}(\mathbf{B}) \setminus \mathbf{0}} \|\mathbf{w}\|$$

Where $\gamma \geq$ and when $\gamma = 1$, it's a non-approximated problem. There are three common variants of SVP [**?**]:

1. Decision (GapSVP$_\gamma$): given a lattice basis $\mathbf{B}$ and a positive integer d, distinguish between the cases $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$ and $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma \cdot d$.
2. Estimate (EstSVP$_\gamma$): given a lattice basis $\mathbf{B}$, compute $\lambda_1(\mathcal{L}(\mathbf{B}))$ up to a γ factor, i.e., output some $d \in [\lambda_1(\mathcal{L}(\mathbf{B})), \gamma \cdot \lambda_1(\mathcal{L}(\mathbf{B}))]$.
3. Search: with is SVP$_\gamma$ itself.

To efficiently compute bounds on the minimum distance, and even find relatively short non-zero lattice vectors, Lenstra-Lenstra-Lovász (LLL) algorithm can be applied. It yields a polynomial-time solution to SVP$_\gamma$ with an approximation factor $\gamma = 2^{n-1} - 2$, which is exponential in the dimension. It "converts an arbitrary lattice into one that generates the same lattice, and which is reduced in the following sense" [**?**].

1.2.2 Shortest Independent Vectors Problem

Approximated Shortest Independent Vector Problem (SIVP$_\gamma$) is: find $\mathbf{U} \in \mathrm{Gl}_n(\mathbb{Z})$ with

$$\|\mathbf{BU}\| \leq \gamma \cdot \min_{\mathbf{V} \in \mathrm{Gl}_n(\mathbb{Z})} \|\mathbf{BV}\|$$

1.2.3 Closet Vector Problem

Approximated Closet Vector Problem (CVP$_\gamma$) is: for $\mathbf{t} \in \mathbb{R}^n$, find a lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that

$$\|\mathbf{v} - \mathbf{t}\| \leq \gamma \cdot \min_{\mathbf{w} \in \mathcal{L}(\mathbf{B})} \|\mathbf{w} - \mathbf{t}\|$$

2 GGH/HNF [?]

Though The GGH/HNF cryptosystem, proposed by Goldreich, Goldwasser and Halevi has already been broken in practice, it's still valuable to learn it while studying lattice based cryptography.

2.1 Related Knowledge

For any $n \times m$ matrices $A \in \mathbb{Z}^{n \times m}$ with rank m, there exists a uni-modular matrix U(i.e. $\det(U) = \pm 1$) such that

$$UA = H,$$

where h_{ii} is positive for $1 \leq i \leq m$, $h_{ij} = 0$ for $j > i$, and $|h_{ij}| < h_{ii}$ for $i > j$ [**?**]. H is called the Hermite normal form of A.

2.2 Key Generation

For a chosen lattice base B, we can calculate its Hermite Normal Form H by finding a uni-modular matrix U(i.e. $\det(U) = \pm 1$) and

$$H = BU.$$

Then B is the private key and H is the public key.

2.3 Encryption

If the message is $m \in \mathbb{Z}^n$, then the ciphertext $c \in \mathbb{Q}^n$ is

$$c = Hm + r,$$

where $r \in \mathbb{Q}^n$ is a small noise vector chosen such that the lattice vector closest to c is Hm.

2.4 Decryption

For the ciphertext message $c \in \mathbb{Q}^n$, the private key B and the public key $H = BU$, first compute:

$$B^{-1} \cdot c = B^{-1} \cdot (Hm + r) = B^{-1}BUm + B^{-1}r = Um + B^{-1}r.$$

Then since r is a small noise by definition, use the babai rounding method [?] to remove the term $B^{-1}r$. Finally get m by

$$U^{-1}Um = m$$

3 NTRU Cryptosystem

The NTRU Cryptosystem is the most practical known lattice-based cryptosystem and arguably the most popular lattice-based cryptosystem. Before we discuss the algorithm details, we need to build up the theoretical set up. And for the sake of simplicity we will omit the mathematical proof and leave only the ready-to-use results.

Definition 2.1: *Let $v \in \mathbb{R}^n$ be a vector and $A \in \mathbb{R}^{n\times n}$ be a matrix. Then, we define A_v^* := (v, Av,..., A^{n-1}v). Furthermore, we define the matrix T, where T is:*

$$\left[\begin{array}{ccc|c} 0 & \cdots & 0 & 1 \\ \hline \ddots & & & 0 \\ & I & & \vdots \\ & & \ddots & 0 \end{array}\right]$$

I is an rotation matrix, hence we have T_v^* is the matrix whose i-th column is equal to v rotated by I

Lemma 2.2: *For any two vectors f, g $\in \mathbb{R}^n$, we have*
(i). $T_f^* g = T_g^* f$
(ii). $T \cdot T_f^* = T_f^* \cdot T$
(iii). $T_f^* \cdot T_g^* = T_{T_f^* g}^*$

Definition 2.3: *A q-ary lattice $\mathcal{L} = \Lambda_q(A)$ is a* ***convolutional modular lattice*** *$A \in \mathbb{Z}^{n\times 2n}$ and $\begin{pmatrix} Tx \\ T_y \end{pmatrix} \in \mathcal{L}$ for all $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{L}$*

Definition 2.4: *Let n, d, $\in \mathbb{N}$ and d ¡ n. A vector f $\in \mathbb{Z}^n$ is called a* ***d-vector*** *if f has exactly d negative and d + 1 positive nonzero entries.*

3.1 NTRU Key Generation

The NTRU cryptosystem requires the generation of both a public and a private key before hand. The formal expression and a mathematical proof of its correctness is provided as follows:

Algorithm 1 NTRU Key Generation

Input: Prime n $\in \mathbb{N}$, modulus q $\in \mathbb{N}$, p $\in \mathbb{N}$ with p ¡ q, weight bound d $\in \mathbb{N}$

Output: A private key $\begin{pmatrix} f \\ g \end{pmatrix} \in \mathbb{Z}^{2n}$ and a public key $h \in \mathbb{Z}_q^n$.

```
1: choose two $d$-vectors $f^{'}, g \in_R \{p, 0, -p\}^n$
2: $f \leftarrow f^{'} + e_1$
3: if $\pi_q\left(T_f^*\right) \notin \mathrm{Gl}_n\left(\mathbb{Z}_q\right)$ then
4:     goto step 1
5: end if
6: h $\leftarrow \left(T_f^*\right)^{-1} g \bmod q$
7: return $\left(\begin{pmatrix} f \\ g \end{pmatrix}, h\right)$
```

Proposition 2.4: *Let* $\left(\begin{pmatrix} f \\ g \end{pmatrix}, h\right)$ *be a key pair generated by NTRU Key Generation Algorithm and* $A := \left(T_f^* T_g^*\right)$. *Then,*

(i). $\Lambda_q(A)$ *is the smallest convolutional modular lattice containing* $\begin{pmatrix} f \\ g \end{pmatrix}$.

(ii). $T_f^* \equiv I(\bmod p)$ *and* $T_g^* \equiv \mathbf{0}(\bmod p)$.

(iii). If $\Lambda_q(A) = \mathcal{L}\left(A'\right)$, *the hermite normal form of* A' *is given by*

$$H = \begin{bmatrix} I & 0 \\ T_h^* & qI \end{bmatrix}$$

Proof: Since we chose $f^{'}, g \in_R \{p, 0, -p\}^n, f^{'}$ and g are divisible by p. As in State 2 in the pseudocode, $f := f^{'} + e_1, f \equiv e_1$ mod p. Since T_f^* and T_g^* are the matrices whose i-th column is equal to f or g rotated by I, the result of congruence shown by (ii) is obvious.

For (i), note that $x \in \Lambda_q(A)$ if there exist $y \in \mathbb{Z}^n$ and $z \in \mathbb{Z}^m$ with $x = A^T y + qz$. Since

$$A^T y = \begin{pmatrix} T_f^* \\ T_g^* \end{pmatrix} y = \sum_{i=0}^{n-1} \begin{pmatrix} T^i f \\ T^i g \end{pmatrix} \cdot y_i \tag{4}$$

and

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \in \mathbb{Z}^m \Leftrightarrow \begin{pmatrix} Tz_1 \\ Tz_2 \end{pmatrix} \in \mathbb{Z}^m \tag{5}$$

Then $\Lambda_q(A)$ is convolutional. Also, if any convolutional modular lattice contains $\begin{pmatrix} f \\ g \end{pmatrix}$, it must contain all $\mathbb{Z}$ -linear combinations of the form (2.1). For (iii), three preliminaries have to be introduced first:

- The vector $\begin{pmatrix} f \\ g \end{pmatrix}$ is contained in $\mathcal{L}(H)$.
- The lattice $\mathcal{L}(H)$ is convolutional.
- The inclusion $\mathcal{L}(H) \subseteq \Lambda_q(A)$ holds.

First, we need $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{Z}^{2n}$ such that $H \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$. Since application of H does not change $x,$ then the x we need becomes $x = f$. For $y := q^{-1}\left(g - T_h^* f\right)$,

$$H \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ T_h^* f + qIy \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{6}$$

By Lemma 2.2 and the definition of h, we have:

$$T_h^* f = T_f^* h = T_f^* \cdot \left(T_f^*\right)^{-1} g \equiv g \pmod{q} \tag{7}$$

Now we have shown that $g - T_h^* f$ is an integer multiple of q, so y is an integral.
From (i), we know that $\Lambda_q(A) \subseteq \mathcal{L}(H)$. Then we need to show that $\mathcal{L}(H)$ is convolutional. We can see that

$$\begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{L}(H) \Leftrightarrow \exists \overline{y} \in \mathbb{Z}^n : H \cdot \begin{pmatrix} x \\ \overline{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \Leftrightarrow T_h^* x + qI\overline{y} = y \tag{8}$$

Let

$$x' := Tx \tag{9}$$

$$y' := q^{-1}\left(Ty - T_h^* x'\right) \tag{10}$$

Then we can calculate that $H \cdot \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Tx \\ Ty \end{pmatrix}$. Then we can calculate

$$Ty \overset{(2.5)}{=} T\left(T_h^* x + qI\overline{y}\right) \equiv TT_h^* x \overset{2.2}{=} T_h^* Tx = T_h^* Tx = T_h^* x' \pmod{q} \tag{11}$$

So, y' is integral, $\mathcal{L}(H)$ is convolutional.
At last, det($\mathcal{L}(H)$) = q^n, and that $\mathcal{L}(H) \supseteq \Lambda_q(A)$, so $\mathcal{L}(H)$ can not contain any further points since both lattices have the same density.

3.2 NTRU Encryption

The following is the psedocode of NTRU encryption algorithm

Algorithm 2 NTRU Encryption

```
Input: Prime n ∈ ℕ, modulus q ∈ ℕ, weight bound d ∈ ℕ, public key $h \in \mathbb{Z}_q^n$, d-vector message
    m ∈ $\{1, 0, -1\}^n$.
Output: Ciphertext $c \in \mathbb{Z}_q^n$.
  choose a d-vector $c \in \mathbb{Z}_q^n$
  h ←$T_f^* g$
  return $m + T_h^* r \bmod q$
```

Definition 2.9: *Let $a, b \in \mathbb{Z}^n$ be two integral vectors and $A \in \mathbb{Z}^{n \times n}$ a matrix of rank: n. We say that a is congruent b modulo A if $A^{-1}(a - b) \in \mathbb{Z}^n$. We then write $a \equiv b$ (mod A)*

The following algorithm helps to reduce a vector modulo a lower triangular matrix A.
Definition 2.10: *Denote the output of Matrix-Reduction by a mod A.*

Lemma 2.11: *For two integral d-vectors $r, m \in \{1, 0, -1\}^n$*

$$\begin{pmatrix} -r \\ m \end{pmatrix} \bmod \begin{pmatrix} I & 0 \\ T_h^* & qI \end{pmatrix} = \begin{pmatrix} 0 \\ (m + T_h^* r) \bmod q \end{pmatrix} \tag{12}$$

Algorithm 3 Matrix Reduction

```
Input: A lower triangular matrix $A = (a_{ij}) \in \mathbb{Z}^{n \times n}$ of rank $n$ and $b \in \mathbb{Z}^n$ .
Output: A vector $\overline{b} \in \mathbb{Z}^n$ such that $0 \leq \overline{b}_i < a_{ij}$ for all $i$ and $b \equiv \overline{b}(\text{mod} A)$.
for $i \in [1, n]$ do
    b $\leftarrow b - \lfloor a_{ii}/b_i \rfloor \cdot (a_{1,i} \cdots a_{n,i})^T$
end for
return b
```

3.3 NTRU Decryption

The psedocode for the decryption of NTRU ciphertext is as follows: We now prove that, With

Algorithm 4 NTRU Decryption

```
Input: prime $n \in \mathbb{N}$, modulus $q \in \mathbb{N}, p \in \mathbb{N}$ with $p < q$, weight bound $d \in \mathbb{N}$, private key
    $\begin{pmatrix} f \\ g \end{pmatrix} \in \mathbb{Z}_q^{2n}$ and a ciphertext $c \in \mathbb{Z}_q^n$.
Output: plaintext $m \in \{1, 0, -1\}^n$
$\overline{v} \leftarrow T_f^* c$
for $i \in [1, n]$ do
    $v_i \leftarrow \underset{\pi_q(v) = \tilde{v}_i}{\arg\min} |v|$
end for
return $(v_1 \cdots v_n) \bmod p$
```

a parameter choice satisfying 8 d p+4 p+2¡q, the NTRU Cryptosystem works correctly.
Proof: Assume that c is a ciphertext generated by NTRU Encryption. Then,

$$T_f^* \cdot c \equiv T_f^* m + T_f^* T_h^* r \overset{3.7}{=} T_f^* m + T_{T_f^* h}^* r \equiv T_f^* m + T_g^* r \quad (\text{mod} q) \tag{13}$$

Hence, let us inspect the vector v more closely. Its i -th entry is given by the formula

$$v_i = \sum_{j=1}^{n} \Big(\left(T_f^*\right)_{ij} m_j + \left(T_g^*\right)_{ij} r_j \Big) = \sum_{j=1}^{n} \left(\left(T^{j-1} f\right)_i m_j + \left(T^{j-1} g\right)_i r_j \right) = \sum_{j=1}^{n} \left(f_{i-j+1} m_j + g_{i-j+1} r_j \right) \tag{14}$$

We write $f' = f - e_1$ as in the NTRU Key Generation step 1 . Estimating the absolute value of v_i, the worst case would certainly be

$$f'_{i-j+1} = \begin{cases} -p; & m_j = -1 \\ p; & m_j = 1 \end{cases} \quad \mathit{And} \quad g_{i-j+1} = \begin{cases} -p & ; \quad r_j = -1 \\ p & ; \quad r_j = 1 \end{cases} \tag{15}$$

since $f = f' + e_1$, we obtain the condition

$$|w_i| \leq (dp + (p+1) + dp) + ((d+1)p + dp) = 4dp + 2p + 1 \tag{16}$$

which yields the condition $8dp + 4p + 2 < q$ if we want the absolute values to be bounded by $q/2$, indicating a successful decryption.

4 The LWE-based cryptosystem

4.1 Introduction

The LWE-based cryptosystem is proved to be the most efficient lattice-based cryptosystem to date which is supported by a theoretical proof of security. The cryptosystem is shown to be secure based on the conjectured hardness of the **L**earning **W**ith **E**rrors problem (LWE), where the definition is shown below.

Algorithm 5 Learning-With-Errors (LWE)

```
*Parameters*: Integers $n, m, q \in \mathbb{Z}_+$ and a probability distribution $\chi : \mathbb{Z}_q \to [0, 1]$.
*Instance*: A pair $(A, v)$ with $A \in \mathbb{Z}_q^{m\times n}$ and $v \in \mathbb{Z}_q^m$.
*Problem Task*: Decide if $v \in_R \mathbb{Z}_q^m$ was chosen uniformly at random or $v = As + e$ was chosen
    with $s \in_R \mathbb{Z}_q^n$ and $e \in_\chi \mathbb{Z}_q^m$.
```

This problem can be equivalently described as a bounded distance decoding problem in q-ary lattices: Given $A \in_R \mathbb{Z}_q^{m\times n}$ and a vector $v \in \mathbb{Z}_q^m$, we need to distinguish between the case that v is chosen uniformly from $\mathbb{Z}_q^m$ and the case in which v is chosen by mangling each coordinate of a random point in $\Lambda_q\left(A^T\right)$ using χ^m.

The LWE problem is believed to be very hard (for reasonable choices of parameters), with the best known algorithms running in exponential time in n. For a real $\alpha > 0$ we let $\chi = \Psi_\alpha$ denote the distribution on $\mathbb{Z}_q$ obtained by sampling a normal variable with mean 0 and standard deviation $\alpha q/\sqrt{2\pi}$, rounding the result to the nearest integer and reducing it modulo q. Furthermore, we have the following theorem proven.

Theorem 1. *Assume access to an oracle that solves the LWE problem with a parameter choice* (n, m, q, χ) *such that* $\chi = \Psi_\alpha,\ \alpha q > \sqrt{n}$, *a prime* $q \leq \mathrm{poly}(n)$ *and* $m \leq \mathrm{poly}(n)$. *Then, there exists a quantum algorithm running in time poly* (n) *for solving the SIVP*$_\gamma$ *and the decision variant of SVP*$_\gamma$ *for* $\gamma = \tilde{O}(n/\alpha)$ *in any lattice of dimension* n.

In other words, when we consider the security of lattice-based problems against quantum computers, any cryptosystem based on LWE is also secure against quantum computers. Moreover, it is very much possible that the proof for Theorem 1 may some day be dequantized, i.e. ported to a classical computational model, leading to a stronger security guarantee for LWE-based cryptosystems. Also, it has to be emphasized that the quantum arguments only show up in the LWE problem itself, and all of the cryptosystems based on it are entirely classical.

4.2 Cryptosystem

4.2.1 Key Generation

The cryptosystem is parameterized by integers n, m, ℓ, t, r, q, and a real $\alpha > 0$. The parameter n is in some sense the main security parameter, and it corresponds to the dimension of the lattices that show up in the worst-case connection. The generation pattern of both the public key and the private key is shown below.

Algorithm 6 LWE-Key-Generation

```
**Input:** $n, m, \ell, t, r, q \in \mathbb{Z}_+$ and $\alpha \in \mathbb{R}_+$
**Output:** private key $S \in \mathbb{Z}_q^{n\times\ell}$ and public key $(A, P) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times\ell}$
1: **choose** $S \in_R \mathbb{Z}_q^{n\times\ell}, A \in_R \mathbb{Z}_q^{m\times n}, E \in \Psi_\alpha\mathbb{Z}_q^{m\times\ell}$
2: **set** $P := AS + E$
3: **return** $(S, (A, P))$
```

4.2.2 Encryption

To better understand the encryption algorithm, the following definition is needed.

Definition 1. *For integers $q, t \in \mathbb{Z}_+$, we define a function $\rho^q_t : \mathbb{Z}_t \to \mathbb{Z}_q$ by*

$$\rho^q_t(n) := \left\lceil \frac{nq}{t} \right\rfloor .$$

We also write ρ^q_t instead of $(\rho^q_t)^\ell$ when we apply this function to vectors in $\mathbb{Z}^\ell_t$.

The encryption algorithm is shown below.

Algorithm 7 LWE-Encryption

```
Input: $n, m, \ell, t, r, q \in \mathbb{Z}_+, \alpha \in \mathbb{R}_+$, public key $(A, P) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times \ell}$, message $v \in \mathbb{Z}_t^\ell$
Output: ciphertext $(u, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^\ell$
choose $a \in R[-r, r]^m \cap \mathbb{Z}^m$
set $u := A^T a$
set $c := P^T a + \rho^q_t(v)$
return $(u, c)$
```

4.2.3 Decryption

The decryption algorithm is shown below. Also, to make the complete cryptosystem more intuitive, a schematic diagram is illustrated in Figure 2 shown below as well.

Algorithm 8 LWE-Decryption

```
Input: ciphertext $(u, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^\ell$, private key $S \in \mathbb{Z}_q^{n\times\ell}$
Output: $v \in \mathbb{Z}_t^\ell$
return $\rho^t_q\left(c - S^T u\right)$
```

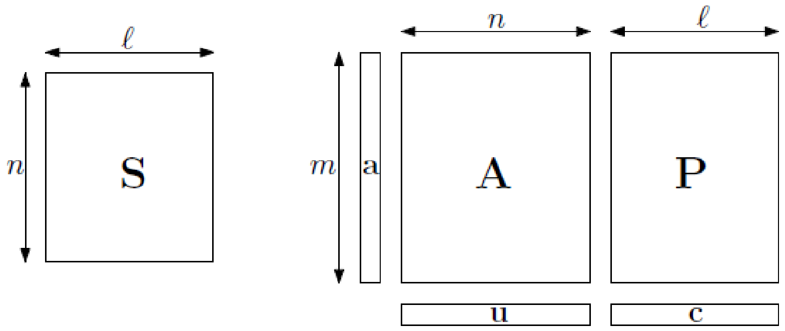

Figure 2: Ingredients in the LWE-based cryptosystem.

4.3 Choosing Parameters

The choice of parameters is meant to guarantee efficiency, a low probability of decryption errors, and security. In the following sections we will discuss how to choose the parameters in detail covering all these aspects.

4.3.1 Efficiency

The cryptosystem can be implemented efficiently since the operations only involve matrix addition and multiplication modulo a certain integer. Because of the features of computers, it is obvious to set $t = 2^k$ to improve the running time. Also, the algorithms can be implemented with high level of parallelization, which is also easy to obtain.

4.3.2 Decryption Errors

Let $b := E^T a$ and assume $|b_i| < \frac{q}{2t} - \frac{1}{2}$. We also set $w := \rho_t^q(v)$, so we know

$$\left|\frac{qv_i}{t} - w_i\right| \leq \frac{1}{2} \Leftrightarrow \left|v_i - \frac{tw_i}{q}\right| \leq \frac{t}{2q}$$

We now get

$$\rho_q^t\left(b_i + \rho_t^q\left(v_i\right)\right) = \left[\frac{t \cdot (b_i + w_i)}{q}\right]$$

Therefore, we can calculate the estimate as

$$\left|v_i - \frac{t \cdot (b_i + w_i)}{q}\right| = \left|v_i - \frac{tw}{q}\right| + \left|\frac{tb_i}{q}\right| < \frac{t}{2q} + \left(\frac{1}{2} - \frac{t}{2q}\right) = \frac{1}{2} \tag{17}$$

Thus, we can get

$$\begin{aligned}\rho_q^t\left(c - S^T u\right) &= \rho_q^t\left(P^T a + \rho_t^q(v) - S^T A^T a\right)\\ &= \rho_q^t\left((AS+E)^T \cdot a + \rho_t^q(v) - S^T A^T a\right)\\ &= \rho_q^t\left(E^T a + \rho_t^q(v)\right) \overset{(17)}{=} v\end{aligned}$$

Since q is odd, if we assume t to be even, we can conclude that

$$|b_i| < \frac{q}{2t} - \frac{1}{2} \Leftrightarrow t\,|b_i| < \frac{q-t}{2} \Leftrightarrow t\,|b_i| < \frac{q}{2} \Leftrightarrow |b_i| < \frac{q}{2t}$$

In other words, under this assumption, we can get

$$\Pr\left[\rho_q^t\left(c - S^T u\right) = v\right] \geq \Pr\left[\forall i : |b_i| < \frac{q}{2t}\right] \tag{18}$$

Now we analyze the behaviour of $b = E^T a$. Since each coordinate of a is uniformly chosen from $[-r, r] \cap \mathbb{Z}$, we can get the variance of each coordinate

$$\operatorname{Var}\left(a_i\right) = \frac{1}{2r+1} \cdot \sum_{k=-r} k^2 = \frac{1}{2r+1} \cdot \frac{r \cdot (r+1) \cdot (2r+1)}{6} = \frac{r(r+1)}{3}$$

We denote by X the random variable measuring b_i and by $Z := \frac{X - \mu(X)}{\sigma(X)}$ its normalization, such that $\Pr\left[z_1 \leq Z \leq z_2\right] = \phi\left(z_2\right) - \phi\left(z_1\right)$. Thus, we can calculate

$$\sigma(X)^2 = \operatorname{Var}\left(b_i\right) = \operatorname{Var}\left(\sum_{j=1}^m E_{ji} a_j\right) = \sum_{j=1}^m \operatorname{Var}\left(E_{ji}\right) \cdot \operatorname{Var}\left(a_j\right) = m \cdot \frac{\alpha^2 q^2}{2\pi} \cdot \frac{r(r+1)}{3}$$

and deduce an upper bound for the decryption error probability per letter:

$$\begin{aligned}\Pr\left[|X| \geq \frac{q}{2t}\right] &= \Pr\left[Z \geq \frac{q}{2t\sigma(X)}\right] + \Pr\left[Z \leq \frac{-q}{2t\sigma(X)}\right]\\ &= 1 - \phi\left(\frac{q}{2t\sigma(X)}\right) + \phi\left(\frac{-q}{2t\sigma(X)}\right)\\ &= 2 - 2 \cdot \phi\left(\frac{q}{2t \cdot \sigma(X)}\right)\\ &= 2 \cdot \left(1 - \phi\left(\frac{1}{2t\alpha} \cdot \sqrt{\frac{6\pi}{r \cdot (r+1) \cdot m}}\right)\right)\end{aligned}$$

Plug into (18), we can obtain

$$\Pr\left[\rho_q^t\left(c - S^T u\right) \neq v\right] < \Pr\left[\forall i : |b_i| \geq \frac{q}{2t}\right] = 1 - \left(1 - \Pr\left[\exists i : |b_i| \geq \frac{q}{2t}\right]\right)^\ell$$

Then, the problem is deduced to mere arithmetics to adjust parameters accordingly to obtain low error margins. With some efforts, we can get the error-correcting codes to the plaintext to reduce the probability of decoding errors to a certain small value, which can be neglected.

4.3.3 Security

To meet the requirements of Theorem 1, we choose q to be prime and $\alpha q > \sqrt{n}$. This leaves us with a choice for m and α where we will attempt to choose α as large as possible since it leads to harder lattice instances. Under these conditions, we may assume that the public keys are completely indistinguishable from pairs (A, P) which is chosen uniformly at random.

Despite all these theoretical results, we would like to describe an attack that is the best among all to break the cryptosystem. Assume that (A, v) is an LWE-instance.

- Choose a short vector $w \in \Lambda_q\left(A^T\right)^*$
- Calculate $\lambda := \langle w, v \rangle$
- If λ is close to an integer, we guess that $v = As + e$ for some $e \in \Psi_\alpha \mathbb{Z}_q^m$

This method is based on the fact that

$$\langle As + e, w \rangle = \underbrace{\langle As, w \rangle}_{\in \mathbb{Z}} + \langle e, w \rangle$$

and relies on $\langle e, w \rangle$ being very small. Fixing any vector w and for $e \in \Psi_\alpha \mathbb{Z}_q^m$,

$$\operatorname{Var}(\langle e, w \rangle) = \operatorname{Var}\left(\sum_{i=1}^m e_i w_i\right) = \sum_{i=1}^m w_i^2 \operatorname{Var}(e_i) = \|a\|^2 \cdot \frac{\alpha^2 q^2}{2\pi}$$

yields a standard deviation of $\|w\| \cdot \frac{\alpha q}{\sqrt{2\pi}}$ for this value. Since we want the above algorithm to fail, we choose

$$\frac{\alpha q}{\sqrt{2\pi}} \gg \frac{1}{\|w\|} \tag{19}$$

Based on the lattice reduction method, we have

$$\Lambda_q\left(A^T\right)^* = q^{-1} \Lambda_q^\perp\left(A^T\right)$$

which result in

$$\|w\| \approx q^{-1} \cdot \min\left(q, 2^{\sqrt{4n \log(q) \log(\delta)}}\right)$$

If right and left hand side of (19) differ by a factor of 1.5, this brings the distribution of λ mod $\mathbb{Z}$ into negligible statistical distance from the uniform distribution. Then, we get

$$\alpha \geq 1.5 \cdot \sqrt{2\pi} \max\left(q^{-1}, 2^{-2}\sqrt{n \log(q) \log(\delta)}\right) \tag{20}$$

For a pair $(A, P) \in_R \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times \ell}$ chosen uniformly at random and used as a public key for LWE-Encryption, we would like the ciphertext to be indistinguishable from a uniformly chosen vector. This would establish the cryptosystem extremely secure against chosen plaintext attacks.

Theorem 2. *Let $r < q$ be integers, $M \in R\mathbb{Z}_q^{n\times m}$ and $a \in R[-r,r]^m \cap \mathbb{Z}_q^m$. Then, the statistical distance between the distribution of Ma and the uniform distribution on $\mathbb{Z}_q^n$ is bounded by*

$$\beta_{n,m}(q,r) := \sqrt{\frac{(2r+1)^m}{q^n}}$$

Choosing $M = \begin{pmatrix} A^T \\ PT \end{pmatrix}$, Theorem 2 suggest we should choose parameters in such a way that $\beta_{n+\ell,m}(q,r)$ is negligible, which, in other words, $\beta_{n+\ell,m}(q,r) = 2^{-100}$.

4.3.4 Summary

In this summary, we present some concrete choices of parameters, satisfying our aforementioned requirements. Thus, we expect them to deliver high levels of both security and efficiency. Equation (19) result in the following choice:

$$m := \frac{(n+\ell)\log(q) + 200}{\log(2r+1)}$$

Using an approximation parameter of $\delta = 1.01$ corresponds to the most exact algorithm known, thus the estimate (20) result in a choice of

$$\alpha := 4 \cdot \max\left\{q^{-1}, 2^{-2\sqrt{n\log(q)\log(1.01)}}\right\}$$

We then can list some properties for the cryptosystem, which are quite intuitive and easy to observe, or otherwise ease to be derived from the above listed equations. The sizes are all in bits, and the logarithms are base 2.

- Private key size: $n\ell \log q$
- Public key size: $m(n+\ell)\log q$
- Message size: $\ell \log t$
- Ciphertext size: $(n+\ell)\log q$
- Encryption blowup factor: $\left(1+\frac{n}{\ell}\right)\log q / \log t$
- Error probability per letter: $2 \cdot \left(1 - \phi\left(\frac{1}{2t\alpha} \cdot \sqrt{\frac{6\pi}{r\cdot(r+1)\cdot m}}\right)\right)$
- Lattice dimension in attack: $\sqrt{n\log(q)/\log(\delta)}$

n	136	166	192	214	233	233
l	136	166	192	214	233	233
m	2008	1319	1500	1333	1042	4536
q	2003	4093	8191	16381	32749	32749
r	1	4	5	12	59	1
t	2	2	4	4	2	40
α	0.0065	0.0024	0.0009959	0.00045	0.000217	0.000217
Public key size in bits	6×10^6	5.25×10^6	7.5×10^6	8×10^6	7.3×10^6	31.7×10^6
Encryption blowup factor	21.9	24	13	14	30	5.6
Error probability	0.9%	0.56%	1%	0.8%	0.9%	0.9%
Lattice dimension in attack	322	372	417	457	493	493

Table 1: Some possible choices of parameters using $\delta = 1.01$.