The Mathematics of the NTRU Public Key Cryptosystem

Abderrahmane Nitaj

The Mathematics of the NTRU Public Key Cryptosystem

Abderrahmane Nitaj
Laboratoire de Mathématiques Nicolas Oresme
Université de Caen Basse Normandie, France

ABSTRACT
The NTRU cryptosystem is a fast public key cryptosystem presented in 1996 by Hoffstein, Pipher and Silverman. It is resistant to quantum attacks and is categorized as a post quantum cryptosystem. In this chapter, we describe the mathematics of the NTRU cryptosystem and the hard problems that make the security of NTRU strong and resistant to classical and quantum attacks.

1 INTRODUCTION

The NTRU public key cryptosystem is one of the fastest known public key cryptosystems. It was first introduced in the rump session at Crypto'96 by Hoffstein, Pipher, and Silverman [Hoffstein et al.,1996], and was later published in the proceedings of the ANTS-III conference. It offers both encryption (NTRU-encrypt) and digital signature (NTRUSign) and is more efficient than the current and more widely used public-key cryptosystems, such as RSA [Rivest et al.,1978], ECC [Koblitz, 1985] [Miller,1985] and El Gamal [El Gamal,1985]. The security of RSA, ECC and El Gamal are based on the difficulty of factoring large composite integers or computing discrete logarithms. In 1997, Shor [Shor,1997] showed that quantum computers can be used to factor integers and to compute discrete logarithms in polynomial time. As a consequence, RSA, ECC and El Gamal will be easily breakable using a quantum computer, and many efforts have been deployed to ensure the future viability of cryptographic protocols in the presence of large scale quantum computers. Hence, some public key cryptosystems have been developed that are believed to be resistant to quantum computing based attacks such as the NTRU cryptosystem. An interesting advantage of NTRU over traditional public-key cryptosystems based on factoring or discrete logarithm is its potential resistance to quantum computers. This makes it a promising alternative to the more established public key cryptosystems. For this reason, NTRU is considered as one of the prominent post quantum cryptosystems. The security of NTRU is related to a very hard problem in lattice reduction, called the shortest vector problem (SVP) and it is conjectured that there is no polynomial time algorithm to solve this problem. On the other hand, the NTRU cryptosystem has been approved for standardization by the Institute of Electrical and Electronics Engineers (IEEE) in 2009.

The mathematics behind the NTRU cryptosystem are intriguing and combine several notions and concepts from algebra, number theory and lattice reduction techniques. In this chapter, we provide an overview of the main theory used to build the NTRU cryptosysem, discuss its classical security as well as its resistance to quantum attacks.

2 DESCRIPTION OF NTRU

2.1 The NTRU encryption scheme

The arithmetic of NTRU depends on three integer parameters (N, p, q). Let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denote the ring of integers modulo q. The operations of NTRU took place in the ring of truncated polynomials $\mathcal{P} = \mathbb{Z}_q[X]/\left(X^N - 1\right)$. In this ring, a polynomial f is defined by its coefficients in the base $\left\{1, X, X^2, \ldots, X^{N-1}\right\}$ as

$$f = (f_0, f_1, \ldots, f_{N-1}) = f_0 + f_1 X + \ldots + f_{N-1} X^{N-1}.$$

The addition of two polynomials f and g is defined as pairwise addition of the coefficients of the same degree

$$f + g = (f_0 + g_0, f_1 + g_1, \ldots, f_{N-1} + g_{N-1}),$$

and multiplication, noted " $*$ " is defined as a convolution multiplication

$$f * g = h = (h_0, h_1, \ldots, h_{N-1}), \text{ with } h_k = \sum_{i+j \equiv k \pmod N} f_i g_j.$$

The Euclidean norm or the length of a polynomial $f = (f_0, f_1, \cdots, f_{N-1})$ is defined as

$$\|f\| = \sqrt{\sum_{i=0}^{N-1} f_i^2}.$$

Let $\mathcal{B}(d))$ be the binary set of polynomials defined for a positive integers d as the set of polynomials of $\mathcal{R}$ with d coefficients equal to 1 and all the other coefficients equal to 0. The set $\mathcal{B}(d))$ can be written as

$$\mathcal{B}(d) = \left\{ f(X) = \sum_{i=0}^{N-1} f_i X^i \in \mathcal{P} \mid f_i \in \{0, 1\}, \sum_{i=0}^{N-1} f_i = d \right\}.$$

Different descriptions of NTRUEncrypt, and different proposed parameter sets, have been in circulation since 1996. The 2005 instantiation of NTRU is set up by six public integers N, p, q, d_f, d_g, d_r and four public spaces $\mathcal{L}_f$, $\mathcal{L}_g$, $\mathcal{L}_m$, $\mathcal{L}_r$.

- N is prime and sufficiently large to prevent lattice attacks.
- p and q are relatively prime numbers.
- q is much larger than p.
- $\mathcal{L}_f = \mathcal{B}(d_f)$ is a set of small polynomials from which the private keys are selected.

- $\mathcal{L}_g = \mathcal{B}(d_g)$ is a similar set of small polynomials from which other private keys are selected.
- $\mathcal{L}_m = \mathbb{Z}_p[X]/\left(X^N - 1\right)$ is the plaintext space. It is a set of polynomials $m \in \mathbb{Z}_p[X]/(X^N - 1)$ that represent encryptable messages.
- $\mathcal{L}_r = \mathcal{B}(d_r)$ is a set of polynomials from which the blinding value used during encryption is selected.

The key generation, encryption and decryption primitives are as follows:

1. **Key generation**
   - Randomly choose a polynomial $f \in \mathcal{L}_f$ such that f is invertible in $\mathcal{P}$ modulo p and modulo q.
   - Compute $f_p \equiv f^{-1} \pmod{p}$ and $f_q \equiv f^{-1} \pmod{q}$.
   - Randomly choose a polynomial $g \in \mathcal{L}_g$.
   - Compute $h \equiv g * f_q \pmod{q}$.
   - Publish the public key (N, h) and the set of parameters p, q, $\mathcal{L}_f$, $\mathcal{L}_g$, $\mathcal{L}_r$ and $\mathcal{L}_m$.
   - Keep the private key (f, f_p).
2. **Encryption**
   - Represent the message as a polynomial $m \in \mathcal{L}_m$.
   - Randomly choose a polynomial $r \in \mathcal{L}_r$.
   - Encrypt m with the public key (N, h) using the rule $e \equiv p * r * h + m \pmod{q}$.
3. **Decryption**
   - The receiver computes $a \equiv f * e \pmod{q}$.
   - Using a centering procedure, transform a to a polynomial with coefficients in the interval $\left[-\frac{q}{2}, \frac{q}{2}\right[$.
   - Compute $m \equiv f_p * a \pmod{p}$.

The decryption process is correct if the polynomial $p * r * g + f * m \pmod{q}$ is actually equal to $p * r * g + f * m \in \mathbb{Z}[X]/\left(X^N - 1\right)$, that is without using modulo q. We have

$$\begin{aligned} a &\equiv f * e \pmod{q} \\ &\equiv f * (p * r * h + m) \pmod{q} \\ &\equiv f * r * (p * g * f_q) + f * m \pmod{q} \\ &\equiv p * r * g * f * f_q + f * m \pmod{q} \\ &\equiv p * r * g + f * m \pmod{q}. \end{aligned}$$

Hence, if $a = p * r * g + f * m$ in $\mathbb{Z}[X]/\left(X^N - 1\right)$, then

$$a * f_p \equiv (p * r * g + f * m) * f_p \equiv m \pmod{p}.$$

We note that if the parameters are chosen properly, the decryption process never fails. A sufficient condition for this is to choose q much larger than p.
We notice that NTRU should not be used without padding because, as explained in [Jaulmes et al.,2000], NTRU is vulnerable to a simple chosen ciphertext attack. To avoid this attack, a padding scheme like NAEP [Howgrave-Graham et al.,2003] should be used.
According to the latest research, the parameters of the following table are considered secure.

Parameters	N	p	q
Moderate security	167	3	128
Standard security	251	3	128
High security	347	3	128
Highest security	503	3	256

Table 1: Security parameters of the NTRU cryptosystem.

2.2 An example of NTRU encryption

To illustrate the NTRU scheme, consider the following parameters

$$\begin{aligned} &N = 11, \\ &p = 3, \\ &p = 61, \\ &f = -X^{10} - X^8 - X^6 + X^4 + X^2 + X + 1, \\ &g = -X^9 - X^8 - X^6 + X^4 + X^2 + 1, \\ &m = X^7 - X^4 + X^3 + X + 1, \\ &r = -X^9 + X^7 + X^4 - X^3 + 1. \end{aligned} \tag{1}$$

Then, we get

$$\begin{aligned} &f_p = X^9 + X^7 + X^5 + 2X^4 + 2 * X^3 + 2X^2 + X, \\ &f_q = 45X^{10} + 49X^9 + 26X^8 + 40X^7 + 53X^6 + 47X^5 + 21X^4 + 24X^3 + 60X^2 + 32X + 31, \\ &h = 11X^{10} + 49X^9 + 25X^8 + 46X^7 + 28X^6 + 53X^5 + 31X^4 + 36X^3 + 32X^2 + 5X + 50, \\ &e = 11X^{10} + 46X^9 + 52X^8 + 35X^7 + 30X^6 + 26X^5 + 35X^4 + 32X^3 + 18X^2 + 56X + 28, \end{aligned} \tag{2}$$

Then, computing $a = f * e \pmod q$ and centering modulo q, we get

$$\begin{aligned} a &= 58X^{10} + 60X^9 + 60X^8 + 4X^7 + 56X^5 + 6X^4 + 55X^2 + 3X + 6, \\ a &= -3X^{10} - X^9 - X^8 + 4X^7 - 5X^5 + 6X^4 - 6X^2 + 3X + 6, \end{aligned} \tag{3}$$

Finally, computing $f_p * a \pmod p$ and centering modulo p, we get

$$\begin{aligned} f_p * a &= X^7 + 2 * X^4 + X^3 + X + 1 \pmod p, \\ f_p * a &= X^7 - X^4 + X^3 + X + 1, \end{aligned} \tag{4}$$

which matches the original message m.

3 LATTICE THEORY

In this section, we will review some concepts of the lattice theory that are useful for this chapter. For more details on lattice theory, we refer to [Micciancio et al.,2002] and [de Weger,2012]. We also describe some classical lattice problems, especially the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) and their connection to cryptography. Finally, we describe the LLL algorithm, which is the main technique in lattice reduction.

3.1 Basic notions on lattices

The LLL algorithm was invented in 1982 and was called LLL after its inventors A.K. Lenstra, H.W. Lenstra et L. Lovász [Lenstra et al.,1982]. Originally, it was aimed to factor polynomials with integer coefficients. Since its invention, the LLL algorithm has served in many topics such as solving diophantine equations and cryptanalysis of certain cryptosystems. It is mainly used to find a very good basis for discrete sets of $\mathbb{R}^n$, called lattices.

Definition 1. Let n and d be two positive integers. Let $b_1 \cdots, b_d \in \mathbb{R}^n$ be d linearly independent vectors. The lattice $\mathcal{L}$ generated by $(b_1 \cdots, b_d)$ is the set

$$\mathcal{L} = \sum_{i=1}^{d} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z} \right\}.$$

The vectors $b_1 \cdots, b_d$ are called a vector basis of $\mathcal{L}$. The lattice rank is n and the lattice dimension is d. If $n = d$ then $\mathcal{L}$ is called a full rank lattice.

If $\mathcal{L} \subset \mathbb{R}^n$ is a lattice of dimension d, then it is an additive subgroup of $\mathbb{R}^n$ and a basis for $\mathcal{L}$ can be written as the rows of a $d \times n$ matrix.

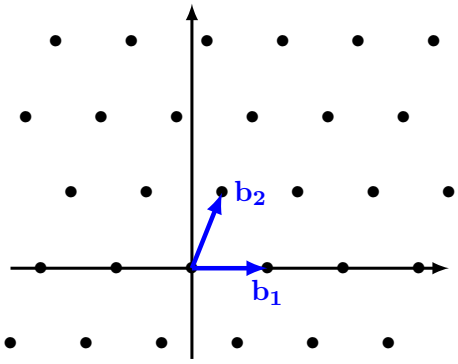

Figure 1: A lattice with the basis (b_1, b_2)

A lattice $\mathcal{L}$ with dimension $d \geq 2$ has infinitely many bases. Any two such bases have the same number of elements and are related with a unimodular matrix.

Theorem 2. *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice of dimension d. Let $(b_1 \cdots, b_d)$ and $(b'_1 \cdots, b'_d)$ be two bases of $\mathcal{L}$. Then there exists a $d \times d$ matrix U with entries in $\mathbb{Z}$ and $\det(U) = \pm 1$ such that*

$$[b'_1, \ldots, b'_d]^t = U[b_1, \ldots, b_d]^t,$$

where v^t is the transpose vector of v.

A lattice has many invariants. An important invariant is the volume or the determinant.

Definition 3. Let $\mathcal{L}$ be a lattice with a basis $(b_1 \cdots, b_d)$. The volume or determinant of $\mathcal{L}$ is

$$\det(\mathcal{L}) = \sqrt{\det(BB^t)},$$

where B is the $d \times n$ matrix formed by the rows of the basis.

Theorem 4. *Let $\mathcal{L}$ be a lattice of dimension d. Then the determinant $\det(\mathcal{L})$ is independent of the choice of the basis.*

When $d = n$, L is called a full-rank lattice, and the matrix of the basis is a $n \times n$ matrix. Then the following property holds.

Theorem 5. *Let $\mathcal{L}$ be a full-rank lattice of dimension n. If $(b_1 \cdots, b_n)$ is a basis of $\mathcal{L}$ with matrix B, then*

$$\det(L) = |\det(B)|.$$

Lattices whose bases have integer coordinates are very convenient for various problems. Such lattices are called *integral lattices*. Also, many problems in lattice theory involve inner product of vectors and distance minimization. The most intuitive way to measure distance in a lattice is by using the Euclidean norm.

Definition 6. Let $u = (u_1, \cdots, u_n)$ and $v = (v_1 \cdots, v_n)$ be two vectors of $\mathbb{R}^n$.

1. The inner product of u and v is
$$\langle u, v \rangle = u^t v = \sum_{i=1}^{n} u_i v_i.$$
2. The Euclidean norm of u is
$$\|u\| = (\langle u, u \rangle)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} u_i^2 \right)^{\frac{1}{2}}.$$

Lattices are used as a fundamental tool for cryptanalysis of various public key cryptosystems such as knapsack cryptosystems, RSA [Rivest et al., 1978], NTRU [Hoffstein et al.,1996] and GGH [Goldreich et al,1997]. On the other hand, lattices are used as a theoretical tool for security analysis of several cryptosystems such as NTRU and LWE [Regev,2005]. These cryptosystems are related to hard computational problems on lattices such the shortest vector problem.

Definition 7. Let L be a lattice. The minimal distance λ_1 of $\mathcal{L}$ is the length of the shortest non-zero vector of $\mathcal{L}$:

$$\lambda_1 = \inf\{\|v\| \in \mathcal{L} \mid v \in \mathcal{L} \backslash \{0\}\}.$$

Another way to define λ_1 is

$$\lambda_1 = \inf\{\|v - u\| \in \mathcal{L} \mid v, u \in \mathcal{L},\ v \neq u\}.$$

Definition 8. Let L be a lattice of dimension n. For $i = 1, \ldots n$, the i-th successive minimum of the lattice is

$$\lambda_i = \min\{\max\{\|v_1\|, \ldots, \|v_i\|\} \mid v_1, \ldots, v_i \in \mathcal{L} \text{ are linearly independent}\}.$$

In the following, we list some computational problems that seem to be hard in general and on which some cryptographic systems have been based. An overview of many hard lattice problems and their interconnections is presented in [LaLaarhoven et al.,2012].

Definition 9. Let $\mathcal{L}$ be a full rank lattice of dimension n in $\mathbb{Z}^n$.

1. **The Shortest Vector Problem (SVP):** Given a basis matrix B for $\mathcal{L}$, compute a non-zero vector $v \in \mathcal{L}$ such that $\|v\|$ is minimal, that is $\|v\| = \lambda_1(\mathcal{L})$.

2. **The Closest Vector Problem (CVP):** Given a basis matrix B for $\mathcal{L}$ and a vector $v \notin \mathcal{L}$, find a vector $u \in \mathcal{L}$ such that $\|v - u\|$ is minimal, that is $\|v - u\| = \mathrm{d}(v, \mathcal{L})$ where $\mathrm{d}(v, \mathcal{L}) = \min_{u \in \mathcal{L}} \|v - u\|$.

3. **The Shortest Independent Vectors Problem (SIVP):** Given a basis matrix B for $\mathcal{L}$, find n linearly independent lattice vectors $v_1, v_2, \ldots, v_n$ such that $\max_i \|v_i\| \leq \lambda_n$, where λ_n is the nth successive minima of $\mathcal{L}$.

4. **The approximate SVP problem (γSVP):** Fix $\gamma > 1$. Given a basis matrix B for $\mathcal{L}$, compute a non-zero vector $v \in \mathcal{L}$ such that $\|v\| \leq \gamma\lambda_1(\mathcal{L})$ where $\lambda_1(\mathcal{L})$ is the minimal Euclidean norm in $\mathcal{L}$.

5. **The approximate CVP problem (γCVP):** Fix $\gamma > 1$. Given a basis matrix B for $\mathcal{L}$ and a vector $v \notin \mathcal{L}$, find a vector $u \in \mathcal{L}$ such that $\|v - u\| \leq \gamma\lambda_1 \mathrm{d}(v, \mathcal{L})$ where $\mathrm{d}(v, \mathcal{L}) = \min_{u \in \mathcal{L}} \|v - u\|$.

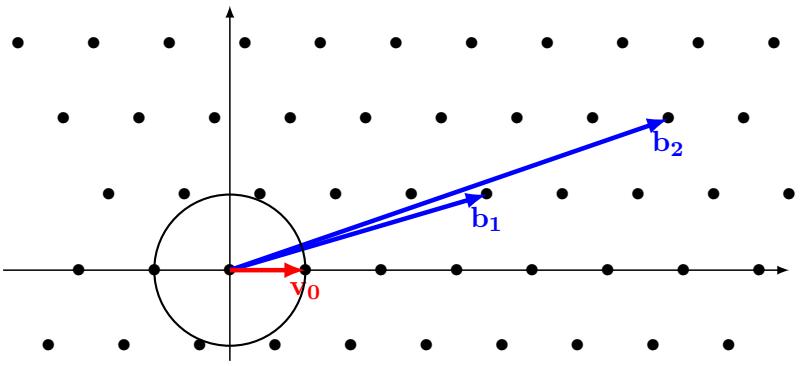

Figure 2: The shortest vectors are v_0 and $-v_0$

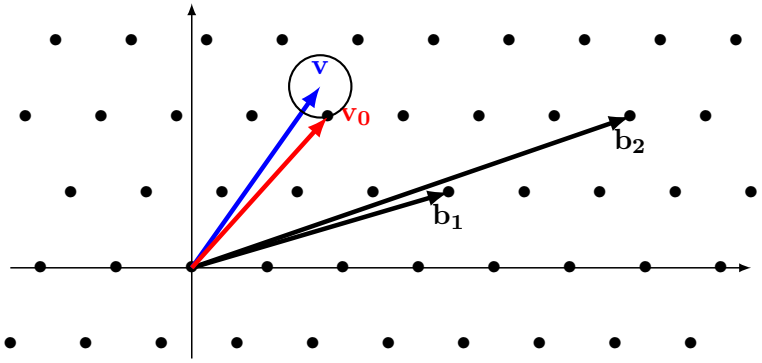

Figure 3: The closest vector to v is v_0

Some of such problems have been shown to be NP-hard, and in general, are known to be hard when the dimension is sufficiently large. No efficient algorithm is known to find the shortest vector nor the closest vector in a lattice. The next result, due to Minkowski gives a theoretical explicit upper bound in terms of $\dim(\mathcal{L})$ and $\det(\mathcal{L})$.

Theorem 10 (Minkowski)**.** *Let $\mathcal{L}$ be a lattice with dimension n. Then there exists a non-zero vector $v \in \mathcal{L}$ satisfying*

$$\|v\| \leq \sqrt{n}\det(L)^{\frac{1}{n}}.$$

On the other hand, the Gaussian Heuristic implies that the expected shortest non-zero vector in a lattice $\mathcal{L}$ is approximately $\sigma(\mathcal{L})$ where

$$\sigma(\mathcal{L}) = \sqrt{\frac{\dim(\mathcal{L})}{2\pi e}}\left(\det(\mathcal{L})\right)^{\frac{1}{\dim(\mathcal{L})}}.$$

We notice that Minkowski's theorem as well as the Gaussian Heuristic are not useful for practical implementations. For implementation purposes, the LLL algorithm is more useful and approximately solves the SVP within a factor of $2^{n/2}$.

3.2 The LLL algorithm

The LLL algorithm is the most useful tool in the algorithmic study of lattices. It provides a partial answer to SVP since it runs in polynomial time and approximates the shortest vector of a lattice of dimension n up to a factor of $2^{n/2}$. On the other hand, Babai [Babai,1986] gave an algorithm that approximates the CVP problem by a factor of $\left(3/\sqrt{2}\right)^n$. In some cases, the LLL algorithm gives extremely striking results both in theory and practice that are enough to solve lattice problems.
The LLL algorithm uses the well known Gram-Schmidt orthogonalization method. The Gram-Schmidt process is an iterative method to orthonormalize the basis of a vector space.

Theorem 11 (Gram-Schmidt)**.** *Let V be a vector space of dimension n and $(b_1 \cdots, b_n)$ a basis of V. Let $(b_1^* \cdots, b_n^*)$ be n vectors such that*

$$b_1^* = b_1, \quad b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*,$$

where, for $j < i$

$$\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}.$$

Then $(b_1^* \cdots, b_n^*)$ *is an orthogonal basis of* V.

Using matrices, the bases $(b_1^* \cdots, b_n^*)$ and $(b_1 \cdots, b_n)$ satisfy

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ \mu_{2,1} & 1 & 0 & 0 & \cdots & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{n_1,1} & \mu_{n-1,2} & \mu_{n-1,3} & \cdots & 1 & 0 \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \cdots & \mu_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} b_1^* \\ b_2^* \\ b_3^* \\ \vdots \\ b_{n-1}^* \\ b_n^* \end{bmatrix}.$$

Clearly the basis $(b_1^* \cdots, b_n^*)$ is an orthogonal basis, but in general, if $(b_1 \cdots, b_n)$ is a basis of a lattice $\mathcal{L}$, $(b_1^* \cdots, b_n^*)$ is not a basis for $\mathcal{L}$.
The Gram-Schmidt process can be transformed into the algorithm shown in 1.

Algorithm 1 : Gram-Schmidt process

INPUT: A basis $(b_1 \cdots, b_n)$ of a space vector $V \subset \mathbb{R}^n$.
OUTPUT: An orthogonal basis $(b_1^* \cdots, b_n^*)$ of V.

1: Set $b_1^* = b_1$.
2: **for** $i = 1, 2, \cdots n$, **do**
3: **for** $j = 1, 2, \cdots i-1$, **do**
4: Compute $\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$.
5: **end for**
6: Compute $b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$.
7: **end for**

The following result shows how to compute the determinant of a lattice with a basis $B = \{b_1, \ldots, b_n\}$ using the Gram-Schmidt orthogonalization.

Corollary 12 (Hadamard)**.** *Let* $B = \{b_1, \ldots, b_n\}$ *be a basis of a lattice* $\mathcal{L}$ *and let* $B^* = \{b_1^*, \ldots, b_n^*\}$ *be the associated Gram-Schmidt orthogonalization. Then*

$$\det(\mathcal{L}) = \prod_{i=1}^{n} \|b_i^*\| \leq \prod_{i=1}^{n} \|b_i\|.$$

The LLL algorithm is connected to the Gram-Schmidt orthogonalization process and produces a basis that satisfies the LLL-reduction notion as in the following definition.

Definition 13. Let $(b_1 \cdots, b_n)$ be a basis of a lattice $\mathcal{L}$. It is said to be LLL-reduced if the Gram-Schmidt orthogonalization $(b_1^* \cdots, b_n^*)$ satisfies

$$|\mu_{i,j}| \leq \frac{1}{2}, \quad \text{for} \quad 1 \leq j < i \leq n, \tag{5}$$

$$\frac{3}{4}\|b_{i-1}^*\|^2 \leq \|b_i^* + \mu_{i,i-1} b_{i-1}^*\|^2, \quad \text{for} \quad 1 < i \leq n. \tag{6}$$

The condition (6) is called Lovász's condition. If $\mu_{i,j} = 0$ for all i and j, then the basis is orthogonal, and consequently is minimal according to Hadamard's inequality as in Corollary 12.
Since a lattice has infinitely many basis, some basis are better than others. A *good basis* is generally a basis with short and almost orthogonal vectors. Consequently, a LLL-reduced basis is a candidate for a good basis.

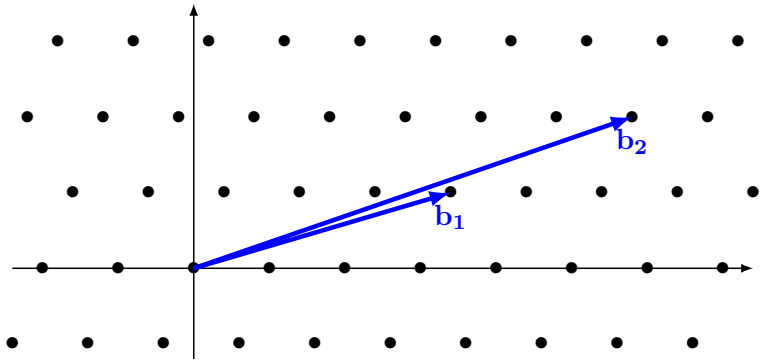

Figure 4: A lattice with *a bad* basis (b_1, b_2)

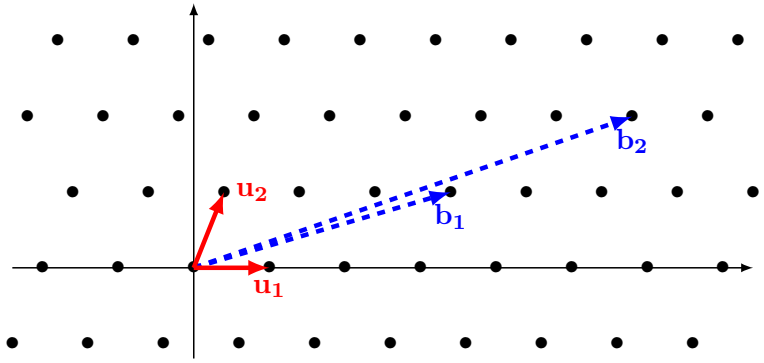

Figure 5: The same lattice with *a good* basis (u_1, u_2)

The original version of the LLL algorithm is presented in Algorithm (2).
An LLL-reduced basis has various properties such as the following ones.

Theorem 14. *Let $(b_1 \cdots, b_n)$ be an LLL-reduced basis with Gram-Schmidt orthogonolization $(b_1^*, \cdots, b_n^*)$. Then*

1. $\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2$ *for* $1 \leq j \leq i \leq n$.
2. $\prod_{i=1}^{n} \|b_i\| \leq 2^{\frac{n(n-1)}{4}} \det(L)$.
3. $\|b_j\| \leq 2^{\frac{i-1}{2}}\|b_i^*\|$ *for* $1 \leq j \leq i \leq n$.

Algorithm 2 : LLL Algorithm

INPUT: A basis $(b_1, \cdots, b_n)$ for $\mathcal{L}$.
OUTPUT: An LLL reduced basis $(b_1, \cdots, b_n)$.

Compute $(b_1^*, \cdots, b_n^*)$ using the Gram-Schmidt orthogonolization method 1.
$k = 2$
while $i \leq n$ **do**
  $b_i = b_i - \sum_{l=1}^{i-1} \lfloor \mu_{i,l} \rceil b_l$
  if $\|b_i^*\|^2 > \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|b_{i-1}^*\|^2$ **then**
    $i = i + 1$
  else
    swap(b_i, b_{i-1})
    $i = \max\{2, i-1\}$
  end if
end while

4. $\|b_1\| \leq 2^{\frac{n-1}{4}} (\det(L))^{\frac{1}{n}}$.

5. *For a non zero vector* $v \in L$, $\|b_1\| \leq 2^{\frac{n-1}{2}} \|v\|$.

The following result fixes the size of the vectors of an LLL-reduced basis.

Theorem 15. *Let* $(b_1, \cdots, b_n)$ *be an LLL-reduced basis. Then for* $1 \leq j \leq n$, *we have*

$$\|b_j\| \leq 2^{\frac{n(n-1)}{4(n-j+1)}} (\det L)^{\frac{1}{n-j+1}}.$$

Note that the LLL algorithm provides a basis of reasonably short vectors and can be used to approximate the shortest vector problem.
The next result shows that the LLL algorithm is a polynomial time algorithm.

Theorem 16. *Let* $(b_1, \cdots, b_n)$ *be a basis of a lattice* $\mathcal{L}$. *Define* $B = \max_i \|b_i\|$. *The LLL algorithm computes an LLL-reduced basis with running time*

$$\mathcal{O}\left(n^4 \log^3 B\right).$$

4 THE LATTICE BASED ATTACK ON NTRU

The NTRU cryptosystem is a polynomial ring cryptosystem and the relation between the public and private key can be used to define a lattice, which is called the NTRU lattice. A basis for this lattice can be derived from the public key, and hence is publicly available. The secret key can be considered as a short vector in this lattice. Consequently, a possible attack on NTRU is to try to solve the approximate shortest vector problem in the NTRU lattice. Indeed, various attack schemes against NTRU have been proposed using lattice reduction [Coppersmith et al.]. On the other hand, different attacks on NTRU have been proposed, without major effects, such as the meet-in-the-middle attacks (see [Howgrave-Graham et

al., 2003] and [Howgrave-Graham,2003]). In the rest of this section, we present the lattice based attack on NTRU presented by Coppersmith and Shamir [Coppersmith et al.,1997] in 1997.
Recall that the NTRU system relies on several parameters, mainly two prime numbers N, q, and an integer p. Also the public key satisfies $h \equiv g * f_q \pmod q$ where g and f are two polynomials of the ring $\mathcal{P}$. Hence $f * h \equiv g \pmod q$. Consider the lattice $\mathcal{L}$ as follows

$$\mathcal{L} = \{(u, v) \in \mathcal{P} \times \mathcal{P} | u * h \equiv v \pmod q\}.$$

Then it is clear that $\mathcal{L} \subset Z^{2N}$ is a lattice and that $(f, g) \in \mathcal{L}$. To find a basis of $\mathcal{L}$, observe that $f * h \equiv g \pmod q$ can be rewritten as $f * h - u * q = g$ for some $u \in \mathcal{P}$. Alternatively, this can be rewritten as

$$\begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \begin{bmatrix} f \\ -u \end{bmatrix}.$$

Using the coordinates of f, g, h and u, this can be transformed into the following form

$$\left[\begin{array}{c} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \\ \hline\hline g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{array}\right] = \left[\begin{array}{cccc||cccc} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ \hline\hline h_0 & h_1 & \cdots & h_{N-1} & q & 0 & \cdots & 0 \\ h_{N-1} & h_0 & \cdots & h_{N-2} & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 & 0 & 0 & \cdots & q \end{array}\right] * \left[\begin{array}{c} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \\ \hline\hline -u_1 \\ -u_2 \\ \vdots \\ -u_{N-1} \end{array}\right].$$

Observe that the matrix is in the form

$$U = \begin{bmatrix} I_N & 0_N \\ M_h & qI_N \end{bmatrix},$$

where I_N is the $N \times N$ identity matrix and M_h is the circulant matrix whose columns are circularly shifted versions of h.
Since $(f, g) \in \mathcal{L}$, then (f, g) is an integer linear combination of the columns of U. Moreover, since the coefficient of (f, g) are small, then (f, g) is a short vector of $\mathcal{L}$, and, with an overwhelming probability, is the shortest vector in $\mathcal{L}$. Consequently, any method that can solve SVP can break the NTRU system. Up to date, there is no efficient way to solve SVP. Alternatively, the LLL algorithm can be applied. Using the matrix U, the LLL algorithm will find a vector b_1 with norm satisfying

$$\|b_1\| \leq 2^{\frac{2N-1}{4}} (\det(L))^{\frac{1}{2N}},$$

while the shortest vector $v \in \mathcal{L}$ satisfies (see Theorem 10)

$$\|v\| \leq \sqrt{2N} \det(L)^{\frac{1}{2N}}.$$

Even if the LLL bound seems very large, in practice, the LLL algorithm outputs a much better bound than the theoretical one. For a small N, the LLL algorithm is sufficient to break the NTRU system as shown by the experiments in [Hoffstein et al.,2003].

5 RESISTANCE TO POST-QUANTUM ATTACKS

In classical physics we have classical bits being either 0 or 1 while, in quantum mechanics, we have qubits. For example, a qubit can be thought of as an electron in a Hydrogen atom with two state system, the ground and the excited state or spin-up and spin-down. Quantum mechanics assert that a two state system can be in any superposition of the two basis states. The state of a qubit can be represented as a vector $|\psi\rangle$ in a two-dimensional vector space with orthonormal basis $\{|0\rangle, |1\rangle\}$ and complex coefficients as shown in Figure 6, so that

$$|\psi\rangle = a|0\rangle + b|1\rangle, \qquad a, b \in \mathbb{C}, \qquad |a|^2 + |b|^2 = 1.$$

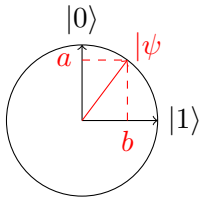

Figure 6: Superposition of the pure states $|0\rangle$ and $|1\rangle$

In column matrix formulation, the basis states are

$$|1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Mathematically, a qubit is a 2-dimensional Hilbert space H_2 so that the state of the qubit is an associated unit length vector in H_2. A qubit can be in state $|0\rangle$ or in state $|1\rangle$ or in a superposition of the two states, that is $a|0\rangle + b|1\rangle$. If a qubit is in state $|0\rangle$ or $|1\rangle$, we say it is a pure state. Otherwise, we say it is a superposition of the pure states $|0\rangle$ and $|1\rangle$.
While the state of a qubit can be represented by a vector in the two dimensional complex vector space H_2, spanned by $|0\rangle$ and $|1\rangle$, a n-qubit system can be represented by a vector in a 2^n-dimensional complex vector space. For $n = 2$, the 2-qubit system corresponds to the tensor product $H_2 \otimes H_2$ which is defined to be the Hilbert space with basis $|i_1\rangle|i_2\rangle$ with $i_1 \in \{0, 1\}$ and $i_2 \in \{0, 1\}$. The possible basis states are $|0\rangle|0\rangle = |00\rangle$, $|0\rangle|1\rangle = |01\rangle$, $|1\rangle|0\rangle = |10\rangle$ and $|1\rangle|1\rangle = |11\rangle$. The basis state $|i_1 i_2\rangle$ means that the first qubit is in its state $|i_1\rangle$ and the second qubit is in its state $|i_2\rangle$. Consider a 2 quantum systems A_1 and A_2, with A_1 in state $\psi_1 = a_1|0\rangle + b_1|1\rangle$ and A_2 in state $\psi_2 = a_2|0\rangle + b_2|1\rangle$. Then the 2 quantum systems is in state

$$\psi_1 \otimes \psi_2 = (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle,$$

with $|a_1a_2|^2 + |a_1b_2|^2 + |b_1a_2|^2 + |b_1b_2|^2 = 1$. Hence, an arbitrary state of a 2 qubit system can be represented by

$$\sum_{i_1 i_2 \in \{0,1\}^2} a_{i_1 i_2}|i_1 i_2\rangle, \quad a_{i_1 i_2} \in \mathbb{C}, \qquad \sum_{i_1 i_2 \in \{0,1\}^2} |a_{i_1 i_2}|^2 = 1.$$

This scheme can be generalized for a n-qubit system. An arbitrary state can be represented by

$$\sum_{i_1 i_2 \ldots i_n \in \{0,1\}^n} a_{i_1 i_2 \ldots i_n} |i_1 i_2 \ldots i_n\rangle, \quad a_{i_1 i_2 \ldots i_n} \in \mathbb{C}, \qquad \sum_{i_1 i_2 \ldots i_n \in \{0,1\}^n} |a_{i_1 i_2 \ldots i_n}|^2 = 1.$$

Hence a n-qubit has 2^n basis states.
In 1997, Shor [Shor,1997] published a quantum algorithm that solves the factorization problem as well as the discrete algorithm problem. The classical part of Shor's algorithm is as in the following Algorithm.

Algorithm 3 : Shor's Algorithm

INPUT: An integer N.
OUTPUT: A non trivial factor of N.

1: If $\gcd(N, 2) = 2$, then return 2.
2: Pick a random integer a with $2 \leq a \leq N - 1$.
3: **if** $\gcd(N, a) = a$ **then**
4: return a.
5: **else**
6: Find the order r of a modulo N, that is the least positive integer r such that $a^r \equiv 1 \pmod{N}$.
7: **if** r is odd **then**
8: go back to step 2.
9: **if** If $a^{r/2} \equiv -1 \pmod{N}$ **then**
10: go back to step 2
11: **else**
12: return $\gcd\left(a^{r/2} - 1 \pmod{N}, N\right)$ and $\gcd\left(a^{r/2} + 1 \pmod{N}, N\right)$.
13: **end if**
14: **end if**
15: **end if**

The quantum part of Shor's algorithm is step 5, which is the periodicity finding technique. As a consequence of Shor's algorithm, classical cryptosystems based on factorization or discrete algorithm problem will be insecure under quantum attacks. The main question this raises is what cryptosystems to use in a quantum world. There are various candidates for a post quantum cryptosystem such as Merkle's hash-tree public-key signature system, McEliece's hidden-Goppa-code public-key system, and the lattice-based cryptosystem NTRU.
In general, lattice problems are quite hard and the best known algorithms either run in exponential time or outputs bad approximations. This is the main motivation for lattice based cryptography. Moreover, lattice problems are believed to resist quantum attacks. Since the discovery of the factorization quantum algorithm by Shor, in 1997, many unfruitful attempts to solve lattice problems by quantum algorithms have been proposed. Hence, it is conjectured that there is no quantum algorithm that solves lattice problems in polynomial time. As a consequence, the NTRU cryptosystem has been categorized as a post-quantum cryptosystem. As noticed above, the quantum part in Shor's algorithm uses periodicity finding technique. For lattice problems, the main difficulty is that the periodicity finding technique does not seem to be applicable. This makes NTRU as one of the promising cryptosystems.

6 CONCLUSION

The NTRU public key cryptosystem, was first presented in 1996 by J. Hoffstein, J. Pipher, and J. H. Silverman and is now included in the IEEE P1363 standard. Comparatively to some classical and well known cryptosystems, such as RSA and ElGamal, the NTRU cryptosystem offers high speed key generation, encryption and decryption. Hence, it can easily be implemented on constrained devices. The security of the NTRU cryptosystem is based on finding a short vector in a lattice of high dimension. This is a very hard problem, even in a quantum word. For these reasons, the NTRU cryptosystem is gaining interest in the electronics industry and makes it a promising alternative for the future of public key cryptography.

REFERENCES

Babai, L. (1986). *On Lovász lattice reduction and the nearest lattice point problem*. Combinatorica, 6, pp. 1–13.

Coppersmith, D. and Shamir, A. (1997). *Lattice attacks on NTRU*. In Advances in cryptology, EUROCRYPT'97, volume 1233 of Lecture Notes in Comput. Sci., pp. 52–61. Springer, Berlin.

El Gamal, T. (1985). *A public key cryptosystem and signature scheme based on discrete logarithms*. IEEE Transactions on Information Theory IT-31, pp. 469–472.

Goldreich, O., Goldwasser S. and Halevi S. (1997). *Public Key Cryptosystems from Lattice Reduction Problems*, Advances in Cryptology-Crypto'97, LNCS 1294 (1997), pp. 112–131.

Hoffstein, J., Silverman, J.H. and Whyte W. (2003). *Estimating breaking times for NTRU lattices*. Technical Report 012, Version 2, 2003.

Howgrave-Graham, N. (2007). *A hybrid lattice-reduction and meet-in-the-middle attack against NTRU*, CRYPTO'07, Springer-Verlag, 2007.

Howgrave-Graham, N., Silverman. J.H. and Whyte W. (2003). *A meet-in-the middle attack on an NTRU private key*, NTRU Cryptosystem Technical Report 004, Version 2, 2003.

Howgrave-Graham N., Silverman J.H., Singer, A. and Whyte, W. (2003). *NAEP: provable security in the presence of decryption failures*. Cryptology ePrint Archive, Report 2003/172, 2003.

Jaulmes, E. and Joux, A. (2000). *A chosen-cipher attack against NTRU*. Lecture Notes in Computer Science, 1880: pp. 20–35, 2000.

Koblitz, N. (1987). *Elliptic curve cryptosystems*, Mathematics of Computation 48, pp. 203–209, 1987.

Laarhoven T., van de Pol, J. and de Weger, B. (2012). *Solving hard lattice problems and the security of lattice-based cryptosystems*, Cryptology ePrint Archive, No. 2012/533, 2012.

Lenstra, A.K., Lenstra, H.W. and Lovász, L. (1982). *Factoring polynomials with rational coefficients*, Mathematische Annalen, Vol. 261, pp. 513–534, 1982.

Miller, V.S. (1985). *Use of elliptic curves in cryptography*, CRYPTO'85, 1985.

Regev, O. (2009). *On lattices, learning with errors, random linear codes, and cryptography*. J. ACM, 6(6):1–40, 2009.

Rivest, R., Shamir A. and Adleman, L. (1978). *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, Vol. 21 (2), 120–126, 1978.

Shor, P.W. (1997). *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Computing 26, pp. 1484–1509, 1997.

ADDITIONAL READING

Diffie W. and Hellman, E. (1976). *New Directions in Cryptography*, IEEE Transactions on Information Theory, 22, 5, pp. 644–654, 1976.

IEEE P1363.1 *Public-Key Cryptographic Techniques Based on Hard Problems over Lattices*, June 2003. IEEE.

McELIECE, R.J. (1978). *A public-key cryptosystem based on algebraic coding theory*, DSN Progress Report 42–44 , Jet Propulsion Laboratory, Pasadena, CA, (1978), pp. 114–116.

Merkle, R. (1989). *A certified digital signature*. In Advances in Cryptology – CRYPTO'89, number 1462 in LNCS, pp. 218–238. Springer, 1989.

Micciancio D. and Goldwasser S. (2002). *Complexity of Lattice Problems, A Cryptographic Perspective*. Kluwer Academic Publishers, 2002.

Parasitism C. and Prada J. (2008). *Evaluation of Performance Characteristics of Cryptosystem Using Text Files*, Journal of Theoretical and Applied Information Technology, Jatit, 2008

KEY TERMS AND DEFINITIONS

Encryption: the process of converting data into a ciphertext, that cannot be understood by unauthorized people.

Decryption: the conversion of encrypted data back into the original form.

Public key cryptosystem: A cryptographic system that uses a public key, known to everyone, and a private or secret key, known only to the recipient of the message.

Quantum computing: the development of computer technology based on the principles of quantum theory.

Quantum computer: a machine based on particles at the sub-atomic level.