Malaysian Journal of Mathematical Sciences 11(S) August: 1 - 28 (2017)
Special Issue: The 5th International Cryptology and Information Security Conference (New Ideas in Cryptology)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

Post Quantum Cryptography

Nitaj, A.

Laboratoire de Mathématiques Nicolas Oresme
Université de Caen Normandie, France

E-mail: abderrahmane.nitaj@unicaen.fr

ABSTRACT

Public key cryptography is widely used for many applications such as signing contracts, electronic voting, encryption, securing transactions over the Internet and storing sensitive data. The discovery of an efficient algorithm based on quantum mechanics for factoring large integers and computing discrete logarithms by Peter Shor in 1994 undermined the security assumptions upon which currently used public key cryptographic algorithms are based, like RSA, El Gamal and ECC. However, some cryptosystems, called post quantum cryptosystems, while not currently in widespread use are believed to be resistant to quantum computing based attacks. In this paper, we provide a survey of quantum and post quantum cryptography. We review the principle of a quatum computer as well as Shor's algorithm and quantum key distribution. Then, we review some cryptosystems undermined by Shor's algorithm as well as some post quantum cryptosystems, that are believed to resist classical and quantum computers.

Keywords: Quantum cryptography, Shor's algorithm, Quantum key distribution, Lattice reduction, LWE cryptosystem, NTRU cryptosystem.

1. Introduction

The purpose of cryptography is to protect the secrets of parties communicating in the presence of adversaries. Many current public key cryptosystems depend upon classical intractable problems, such as factoring large integers and solving the discrete logarithm. In 1994, Shor discovered a very important algorithm that would efficiently solve very hard problems if applied with a quantum computer. This shows that quantum computing will deliver unbelievable performance compared to classical computers. A typical example is the factorization of integers problem. While this problem is believed hard for classical computers, Shor's algorithm can solve this type of problem relatively easily with linear time with a quantum computer. So far, the best classical algorithm for factoring is the number field sieve (Buhler et al., 1994), which runs in sub-exponential time $\mathcal{O}\left(\exp\left(c(\log n)^{1/3}(\log\log n)^{2/3}\right)\right)$ for some constant c. In contrast Shor's algorithm runs in time $O\left((\log n)^2(\log\log n)(\log\log\log n)\right)$ on a quantum computer, and then must perform $\mathcal{O}(\log n)$ steps of post processing on a classical computer. Shor's algorithm encouraged the design and construction of quantum computers and was a motivator for the study of new quantum computer algorithms and new cryptosystems that are secure from quantum computers, called post-quantum cryptosystems.

In this paper, we will consider two kinds of cryptography: classical cryptography and quantum cryptography. Classical cryptography has many applications such as secure communication, identification and authentication, key exchange, digital signatures, data integrity, electronic voting, electronic funds transfer, electronic commerce, certification authority, zero-knowledge and secret sharing. The security of all these applications are based on some often believable hard problems. Typical examples of such hard problems are number theoretic problems. In the present days, two major families of cryptographic primitives dominate public key cryptography.

1. Primitives whose security is believed to be based on the difficulty of the integer factorization problem. Typical examples are
   - The RSA cryptosystem (Rivest et al., 1978) and a series of variants.
   - The Rabin cryptosystem (Rabin, 1979).
   - The KMOV Cryptosystem (Koyama et al., 1991).
2. Primitives whose security is believed to be based on the difficulty of the discrete logarithm problem such as
   - The Diffie-Hellman key exchange (Diffie and Hellman, 1976).

- The ElGamal cryptosystem (ElGamal, 1985).
- The Digital Signature Algorithm (DSA) (FIPS, 2000).
- The elliptic curve cryptography (ECC) (Hoffstein et al., 1998) and (Miller, 1985).

However, advances in quantum computers threaten to undermine most of these security assumptions. Therefore, cryptographers have been led to investigate other mathematical problems to see if they can be applied in cryptography. This makes post-quantum cryptography an important topic of research.

The organization of the paper is as follows. In Section 2, we review the principle of quantum computers. In Section 3, we review two quantum algorithms, Shor's algorithm for factorization and the BB84 protocol for key distribution. In Section 4, we present two cryptosystems that are vulnerable to quantum computers: RSA and ElGamal. In Section 5, we review some hard problems in the theory of lattices upon which the security of some post quantum schemes is based. In Section 5, we present two prominent post quantum cryptosystems, namely NTRU and LWE. We conclude the paper in Section 7.

2. Quantum Computers

In this section we present a basic overview of a quantum computer.

2.1 Qubits

While classical computers operate on bits, a quantum computer operates on qubits, or quantum bits. In physics, a qubit can be thought as one of the systems presented in Table 1.

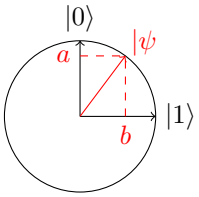
System	**Qubit state**
Electron	Spin
Photon	Polarization

Table 1: Examples of physical 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 complexe coefficients:

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



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 $|1\rangle$.

A classical bit can only be in one of two states, 0 or 1, but a qubit can be in any superposition state. However a measurement of a bit will reveal the bit with probability 1 and will not change the bit. Comparatively, a measurement of a qubit in the state $a|0\rangle + b|1\rangle$ will yield $|0\rangle$ with probability $|a|^2$ or $|1\rangle$ with probability $|b|^2$. After measurement, the state will definitely be $|0\rangle$ or $|1\rangle$. Hence a measurement of a qubit will irreversibly destroy the superposition. Another important property is that a qubit can not be cloned. This property is called the no-cloning theorem.

2.2 Multiple Qubits

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 $n = 2$, a 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

$|1\rangle|1\rangle = |11\rangle$. In column matrix formulation, the basis states are

$$|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \quad |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \quad |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \quad |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

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 and A_2 in state $\psi_2 = a_2|0\rangle + b_2|1\rangle$. Then the 2-quantum system 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.

It happens that some multiple qubits can not be represented as the product of two qubits in the Hilbert space such as $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Indeed, $|\psi\rangle \neq |a\rangle|b\rangle$ for all single qubits $|a\rangle$ and $|b\rangle$. A state of a composite system having this property is called an entangled state.

A quantum computer is a device that use quantum-mechanical phenomena, such as superposition and entanglement. The operations involved in a quantum computer are based on quantum computation, that is transformation of quantum states.

3. Quantum Cryptography

In this section, we present two quantum algorithms, Shor's famous polynomial time quantum algorithm for factoring integers and the BB84 protocol for key distribution.

3.1 Shor's Algorithm

In 1994, Shor (1999) proposed an algorithm on quantum computers for solving the integer factorization problem. Shor also proposed an efficient quantum algorithm for solving the discrete logarithm problem. This illustrates that quantum adversaries would break most of the widely used cryptosystems. The factoring algorithm uses a well known reduction of the factoring problem to the problem of finding the period of a certain function, and it uses the quantum Fourier transform to find the period, which is infeasible with classical computers.

Shor's algorithms have potentially important implications for many cryptosystems when their security is based on the assumption that factoring large numbers is difficult or on the difficulty of computing discrete logarithms. Shor's algorithm consists of two parts: a classical and a quantum part.

The classical part of Shor's algorithm is as follows.

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 $2 \leq a \leq N - 1$.
   (a) If $\gcd(N, a) = a$, then return a. This may be done using the Euclidean algorithm.
   (b) Find the order r of a modulo N, that is the least positive integer r such that $a^r \equiv 1 \pmod{N}$.
      i. If r is odd, then go back to step 2.
      ii. If $a^{r/2} \equiv -1 \pmod{N}$, then go back to step 2.
      iii. Else return $\gcd\left(a^{r/2} - 1 \pmod{N}, N\right)$
      iv. and $\gcd\left(a^{r/2} + 1 \pmod{N}, N\right)$.

The quantum part of Shor's algorithm is as follows.

Input: A composite integer N and an integer a with $2 \leq a \leq N - 1$.

Output: The order r of a modulo N.

1. Find a number $Q = 2^t$ such that $N^2 \leq Q < 2N^2$.

2. Start with a pair of input and output qubit registers with t qubits each, and initialize them to the state

$$|0\rangle|0\rangle = |00\ldots00\rangle|00\ldots00\rangle.$$

3. Apply a Hadamard gate to each qubit in the first register to obtain the state

$$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|0\rangle$$

This state represents a uniform superposition of all the computational basis states in the first register.

4. For each number x in the first register, calculate the quantity $a^x \pmod N$ and store the result in the second register. This produces the following state

$$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|a^x \pmod N\rangle.$$

5. Measure the state of the second register. This reveals a particular value $|a^{x_0} \pmod N\rangle$ for the contents of the second register for some smallest value x_0, and simultaneously projects the state of the first register into a superposition values of $|x_0 + br\rangle$ with $x_0 \leq x_0 + br < Q$, that is

$$0 \leq b \leq \left\lfloor \frac{Q - x_0}{r} \right\rfloor,$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x. Define

$$M = \left\lfloor \frac{Q - x_0}{r} \right\rfloor + 1.$$

Thus the new state is

$$\frac{1}{\sqrt{M}} \sum_{b=0}^{M-1} |x_0 + br\rangle|a^{x_0} \pmod N\rangle.$$

6. Apply the quantum Fourier transform to the first register. The quantum Fourier transform takes the state $|x\rangle$ to the state

$$\frac{1}{\sqrt{Q}} \sum_{y=0}^{Q-1} e^{2\pi i y x/Q} |y\rangle.$$

Hence, the quantum Fourier transform changes the state

$$\frac{1}{\sqrt{M}} \sum_{b=0}^{M-1} |x_0 + br\rangle |a^{x_0} \pmod N\rangle.$$

to the state

$$\begin{aligned}
&\frac{1}{\sqrt{M}} \sum_{b=0}^{M-1} \left(\frac{1}{\sqrt{Q}} \sum_{y=0}^{Q-1} e^{2\pi i y(x_0+br)/Q} |y\rangle \right) |a^{x_0} \pmod N\rangle \\
&= \frac{1}{\sqrt{MQ}} \sum_{b=0}^{M-1} \sum_{y=0}^{Q-1} e^{2\pi i y(x_0+br)/Q} |y\rangle |a^{x_0} \pmod N\rangle \\
&= \frac{1}{\sqrt{MQ}} \sum_{b=0}^{M-1} \sum_{y=0}^{Q-1} e^{2\pi i y x_0/Q} e^{2\pi i y b r/Q} |y\rangle |a^{x_0} \pmod N\rangle \\
&= \frac{1}{\sqrt{MQ}} \sum_{y=0}^{Q-1} e^{2\pi i y x_0/Q} \left(\sum_{b=0}^{M-1} e^{2\pi i y b r/Q} \right) |y\rangle |a^{x_0} \pmod N\rangle.
\end{aligned}$$

7. Perform a measurement on the first register. This yields the state $|y\rangle$ with probability

$$\begin{aligned}
\text{Prob}(y) &= \frac{1}{MQ} \left| \sum_{b=0}^{M-1} e^{2\pi i y b r/Q} \right|^2 \\
&= \frac{1}{MQ} \left| \frac{1 - e^{2\pi i y M r/Q}}{1 - e^{2\pi i y r/Q}} \right|^2 \\
&= \frac{1}{MQ} \frac{\sin^2\left(\frac{\pi y M r}{Q}\right)}{\sin^2\left(\frac{\pi y r}{Q}\right)}.
\end{aligned}$$

8. Define m to be the closest integer to $\frac{yr}{Q}$, that is

$$m = \left\lceil \frac{yr}{Q} \right\rfloor.$$

Then

$$\text{Prob}(y) = \frac{1}{MQ} \frac{\sin^2\left(\pi M \left(\frac{yr-mQ}{Q} + m\right)\right)}{\sin^2\left(\pi \left(\frac{yr-mQ}{Q} + m\right)\right)} = \frac{1}{MQ} \frac{\sin^2\left(\pi M \frac{yr-mQ}{Q}\right)}{\sin^2\left(\pi \frac{yr-mQ}{Q}\right)}.$$

Observe that $\text{Prob}(y)$ is as higher as $|yr - mQ|$ is small. Indeed,

$$\lim_{|yr-mQ|\to 0} \text{Prob}(y) = \lim_{|yr-mQ|\to 0} \frac{1}{MQ} \frac{\sin^2\left(\pi M \frac{yr-mQ}{Q}\right)}{\sin^2\left(\pi \frac{yr-mQ}{Q}\right)} = \frac{M}{Q}.$$

Suppose that $|yr - mQ| \leq \frac{Q}{2r}$. Then

$$\left|\frac{y}{Q} - \frac{m}{r}\right| \leq \frac{1}{2r^2}.$$

It follows that $\frac{m}{r}$, in lowest terms, is a convergent of the continued fraction expansion of $\frac{y}{Q}$. Consequently, the probability $\text{Prob}(y)$ is large when $\frac{m}{r}$ is computed from $\frac{y}{Q}$ by the continued fraction algorithm.

9. Compute the convergents of $\frac{y}{Q}$. Let $\frac{m}{r}$ be a convergent with $r < N$. This procedure yields r if m and r are coprime, but it fails if m and r have any common factors.

10. If $y^r \not\equiv 1 \pmod{N}$, then return to Step 1. On average, this procedure outputs the correct order r in $\log N$ number of repetitions for large N.

Shor showed that the quantum part runs in time $O\left((\log n)^2(\log\log n)(\log\log\log n)\right)$ on a quantum computer, and then must perform $\mathcal{O}(\log n)$ steps of post processing on a classical computer to execute the continued fraction algorithm.

3.2 Quantum Key Distribution

Since the negative impact on public-key cryptography of Shor's algorithms, quantum cryptography has been developed from several points of view. Prior to Shor's work, Bennett and Brassard (1984) proposed in 1984 a quantum key distribution scheme using quantum communication, called BB84. It concerns three main characters, A and B, who try to share a secret key, and E, whose objective is to obtain some information about the secret key. A and B have access to a quantum channel as well as a classical channel. We suppose that E has full access to the quantum channel but it is impossible for him to modify the information sent through the classical channel. According to quantum mechanic principles, it is impossible to duplicate the quantum information. A sends single particles to B across the quantum channel. The particles are produced in two different orthonormal bases, e.g. the rectilinear basis $\{|0\rangle_+, |1\rangle_+\}$ and the diagonal basis $\{|0\rangle_\times, |1\rangle_\times\}$ where

$$|1\rangle_\times = \frac{1}{\sqrt{2}}\left(|0\rangle_+ + |1\rangle_+\right), \quad |0\rangle_\times = \frac{1}{\sqrt{2}}\left(|0\rangle_+ - |1\rangle_+\right).$$

In the BB84 protocol, to exchange a secret key, A and B must proceed as follows.

1. To send a sequence of n bits to B, A encodes each bit in the quantum state of a photon as in Table 2: each bit is encoded in a random basis among the two bases.

	0	1
Basis $\oplus = \{\|\uparrow\rangle, \|\rightarrow\rangle\}$	$\uparrow$	$\rightarrow$
Basis $\otimes = \{\|\nwarrow\rangle, \|\nearrow\rangle\}$	$\nwarrow$	$\nearrow$

Table 2: Encoding bits

   Then, A sends the n photons to B, each in one of the states $|\rightarrow\rangle$, $|\uparrow\rangle$, $|\nearrow\rangle$ or $|\nwarrow\rangle$.

2. For each photon that B receives, he randomly chooses a basis among $\{|\uparrow\rangle, |\rightarrow\rangle\}$ and $\{|\nwarrow\rangle, |\nearrow\rangle\}$ and measures the qubit with respect to the basis.

3. B informs A the basis he used via a classical authentication channel.

4. A checks whether his basis coincides with the basis he received. When both basis coincide, A keeps the corresponding bit.

5. A tells B which bases were correct.

6. A and B can reconstitute a part of the random bit string created previously by A. Statistically, the bases of A and B coincide in 50% of all cases, and the measurements of B agree with the bits of A perfectly. Hence A and B continue with approximately $n/2$ outcomes for which the same basis was used.

7. A and B verify measurement outcomes on random approximately $n/4$ bits of the $n/2$ common bits. Hence, any attempt by E will be detected since E can not copy the qubits and any measurement of a qubit will disturb the state.

8. A and B obtain a common secret key from the remaining about $n/4$ bits.

The BB84 protocol can be shown as in the following example.

A's bits	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1
A's bases	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\oplus$	$\oplus$	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
A's polarizations	$\nwarrow$	$\uparrow$	$\nearrow$	$\nwarrow$	$\nwarrow$	$\rightarrow$	$\uparrow$	$\uparrow$	$\nwarrow$	$\nwarrow$	$\nwarrow$	$\uparrow$	$\uparrow$	$\rightarrow$	$\rightarrow$
B's bases	$\oplus$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\otimes$	$\oplus$	$\otimes$	$\oplus$
B's measurements	$\uparrow$	$\uparrow$	$\uparrow$	$\nwarrow$	$\uparrow$	$\rightarrow$	$\uparrow$	$\uparrow$	$\rightarrow$	$\nwarrow$	$\rightarrow$	$\nwarrow$	$\uparrow$	$\nearrow$	$\rightarrow$
B's bits	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1
Comparison of bases	$\neq$	$=$	$\neq$	$=$	$\neq$	$=$	$=$	$=$	$\neq$	$=$	$\neq$	$\neq$	$=$	$\neq$	$=$
Shared secret bits		0		0		1	0	0		0			0		1

Table 3: A BB84 simulation

4. Cryptosystems Vulnerable to Quantum Computers

Factorization and the discrete logarithm problem have been by far the most productive hard problems in cryptography. These problems will not be difficult if Shor's algorithm is implemented in quantum computers. Consequently, some of the popular cryptosystems will not resist to quantum computers. Table 1 shows some of these cryptosystems as well as the underlying hard problems.

System	Underlying hard problem
RSA	Factorization
Rabin's cryptosystem	Factorization
KMOV	Factorization
Diffie-Hellman key exchange	Discrete Logarithm Problem
El Gamal	Discrete Logarithm Problem
Elliptic Curve Cryptography (ECC)	Elliptic Curve Discrete Logarithm Problem
Digital Signature Algorithm (DSA)	Discrete Logarithm Problem
Elliptic Curve Digital Signature Algorithm (ECDSA)	Elliptic Curve Discrete Logarithm Problem

Table 4: Cryptosystems broken by Shor's algorithm

4.1 Cryptosystems Based on Factorization: RSA

Factoring is the underlying presumably hard problem upon which several public-key cryptosystems are based. This includes RSARivest et al. (1978), Rabin's cryptosystem (Rabin, 1979), LUC (Smith and Lennon, 1993) and KMOV (Koyama et al., 1991).

Factorization: Given a positive integer n, find its prime factorization, that is write $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ where the p_i are pairwise distinct primes and each e_i is a positive integer.

Factoring is widely believed to be a hard problem and the best algorithm for solving it is the Number Field Sieve with a sub-exponential running time. The principal threat comes from a quantum computer on which factoring can be solved efficiently using Shor's algorithm. The most popular cryptosystem based on factorization is RSA. RSA was invented by Rivest, Shamir and Adleman in 1978. It can be summarized as follows:

1. **Key generation**
   - Choose two large primes p and q and compute the RSA modulus $N = pq$.
   - Choose an integer e that is coprime to $(p-1)(q-1)$.
   - Compute d using $ed \equiv 1 \pmod{(p-1)(q-1)}$.
   - Publish the public key (N, e) and keep the private key (N, d).
2. **Encryption**
   - Represent the message to be transmitted as a positive integer $m < N$.
   - Encrypt m with the public key (N, e) using $c \equiv m^e \pmod{N}$.
3. **Decryption**
   - The receiver decrypts the message using $m \equiv c^d \pmod{N}$.
   - Transform the positive integer m into the original message.

The idea of breaking RSA with a quantum computer using Shor's algorithm was a powerful motivator for the design and construction of quantum computers and for the study of new quantum computer algorithms and cryptosystems that are secure from quantum computers.

4.2 Cryptosystems Based on Discrete Logarithms: ElGamal

In 1985, El Gamal described a cryptosystem based on the difficultly of finding a solution to the discrete logarithm in a finite field $\mathbb{F}_p = (\mathbb{Z}/p\mathbb{Z})$ where p is a prime number.

DLP: Given a primitive element g of $\mathbb{F}_p$ and another element a of $\mathbb{F}_p$, the discrete logarithm problem (DLP) is the computational problem of finding x such that $a \equiv g^x \pmod{p}$.

The ElGamal cryptosystem can be summarized as follows:

1. **Key generation**
   - Choose a large prime p and a generator g of the group $(\mathbb{Z}/p\mathbb{Z})^*$.
   - Randomly choose an integer a with $2 \leq a \leq p-2$.

- Compute $b \equiv g^a \pmod{p}$.
- Publish the public key (p, g, b) and keep the private key a.

2. **Encryption**

- Represent the message to be transmitted as a positive integer $m < p$.
- Randomly choose an integer k with $2 \leq k \leq p-2$.
- Encrypt m with the public key (p, g, b) using the rule

$$\gamma \equiv g^k \pmod{p}, \qquad \delta \equiv mb^k \pmod{p}.$$

3. **Decryption**

- The receiver decrypts the message using the rule $m \equiv \gamma^{-a}\delta \pmod{p}$.
- Transform the positive integer m into the original message.

The correctness of the decryption in the ElGamal cryptosystem is as follows. We have

$$\gamma^{-a}\delta \equiv \left(g^k\right)^{-a} mb^k \equiv \left(g^k\right)^{-a} m\left(g^a\right)^k \equiv m \pmod{p}.$$

The main known attack on an ElGamal cryptosystem is to solve the discrete logarithm problem. There are three basic types of discrete logarithm algorithm solvers: Pollard's rho algorithm, the Pohlig-Hellman algorithm, and the index calculus algorithm. The complexity of Pollard's rho algorithm and the Pohlig-Hellman algorithm are exponential while the expected running time of the index calculus algorithm is $O\left(\exp\left(c\sqrt{\log n \log\log n}\right)\right)$ with a constant $c > 0$. For comparison, the running time of Shor's algorithm for discrete logarithm on a quantum computer is $O\left((\log n)^2(\log\log n)(\log\log\log n)\right)$.

5. Lattices

In this section, we introduce the theory of lattices and study some of the most known hard problems in this theory. We start with some notation that is useful for the rest of this paper. Afterwards, we present some important properties of lattices. Finally, we present hard problems in lattices that are used in cryptography.

5.1 Notation

Let n be a positive integer and $\mathbb{R}^n$ be the n-dimensional Euclidean vector space. Let $u = (x_1, \ldots, x_n)$ and $v = (y_1, \ldots, y_n)$ be two vectors of $\mathbb{R}^n$. The inner product of the vectors u, v is

$$\langle u, v \rangle = \sum_{i=1}^{n} x_i y_i.$$

For a vector $u = (x_1, \ldots, x_n)$, the Euclidean norm is defined as

$$\|u\| = \sqrt{\langle u, u \rangle} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

The distance of two vectors $u = (x_1, \ldots, x_n)$ and $v = (y_1, \ldots, y_n)$ is defined as

$$\|u - v\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

5.2 Lattices

Definition 5.1. *Let n and d be two positive integers with $d \leq n$. 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 set of vectors $b_1 \cdots, b_d$ is 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 a discrete additive subgroup of $\mathbb{R}^n$.

Proposition 5.1. *Let $\mathcal{L}$ be a lattice of dimension d and rank n. Then $\mathcal{L}$ can be written as the rows of an $n \times d$ matrix with real entries.*

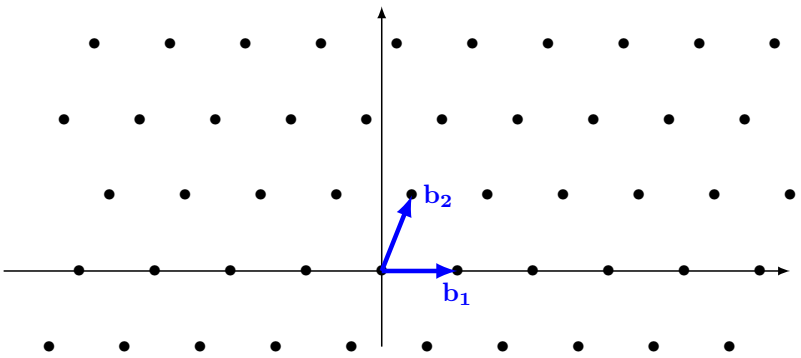


Figure 1: A lattice with the basis $b_1 = (1, 0)$, $b_2 = (0.3, 1)$

Proof. Let $(b_1 \cdots, b_d)$ be a basis of $\mathcal{L}$ such that, for $1 \leq i \leq d$,

$$b_i = \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{ni} \end{bmatrix}.$$

Let v be a vector of $\mathcal{L}$. Then $v = \sum_{i=1}^{d} x_i b_i$ for $x_i \in \mathbb{Z}$. Hence v can be rewritten as

$$\begin{aligned} v &= x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{bmatrix} + \ldots + x_d \begin{bmatrix} a_{1d} \\ a_{2d} \\ \vdots \\ a_{nd} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}. \end{aligned}$$

The involved matrix is constructed using the coordinates of the basis $(b_1 \cdots, b_d)$ as follows

$$\begin{bmatrix} b_1 & b_2 & \cdots & b_d \\ \downarrow & \downarrow & \cdots & \downarrow \\ a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nd} \end{bmatrix}.$$

□

The following result shows that in a lattice $\mathcal{L}$ with dimension $d \geq 2$, any two couples of bases are related by a unimodular matrix.

Proposition 5.2. *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*

$$\begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_d \end{bmatrix} = U \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix}.$$

Proof. Let $(b_1 \cdots, b_d)$ and $(b'_1 \cdots, b'_d)$ be two bases of $\mathcal{L}$. Since every vector $b'_i \in \mathcal{L}$, then

$$\begin{aligned} \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_d \end{bmatrix} &= \begin{bmatrix} u_{11}b_1 + u_{12}b_2 + \ldots + u_{1d}b_d \\ u_{21}b_1 + u_{22}b_2 + \ldots + u_{2d}b_d \\ \vdots \\ u_{d1}b_1 + u_{d2}b_2 + \ldots + u_{dd}b_d \end{bmatrix} \\ &= \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1d} \\ u_{21} & u_{22} & \cdots & u_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ u_{d1} & u_{d2} & \cdots & u_{dd} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix} \\ &= U \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix}, \end{aligned}$$

where U is a $d \times d$ matrix with integer entries. This can be rewritten as

$$(b'_1, b'_2, \ldots, b'_d)^t = U(b_1, b_2, \ldots, b_d)^t.$$

Similarly, there exist a $d \times d$ matrix U' with integer entries such that

$$(b_1, b_2, \ldots, b_d)^t = U'(b'_1, b'_2, \ldots, b'_d)^t.$$

Hence,

$$(b'_1, b'_2, \ldots, b'_d)^t = UU'(b'_1, b'_2, \ldots, b'_d)^t,$$

which implies $UU' = I_d$ where I_d is the $d \times d$ identity matrix. Taking determinant, we get $\det(U)\det(U') = 1$. Since the entries of U and U' are integers, then $\det(U), \det(U') \in \mathbb{Z}$ and $\det(U) = \det(U') = \pm 1$. □

Observe that any $d \times d$ triangular matrix U with diagonal entries equal to ± 1 satisfies $\det(U) = \pm 1$. This shows that a lattice $\mathcal{L}$ with dimension $d \geq 2$ has infinitely many bases.

Definition 5.2. *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 of formed by the rows of the basis.

Proposition 5.3. *Let $\mathcal{L}$ be a lattice of dimension d. Then the $\det(\mathcal{L})$ is independent of the choice of the basis.*

Proof. Let $(b_1 \cdots, b_d)$ and $(b'_1 \cdots, b'_d)$ be two bases of $\mathcal{L}$ with matrices B and B'. Then there exists a $d \times d$ matrix U with entries in $\mathbb{Z}$ and $\det(U) = \pm 1$ such that $B' = UB$. Then since $B'B'^t = UBB^tU^t$, we get

$$\det(B'B'^t) = \det\left(UBB^tU^t\right) = \det(U)\det\left(BB^t\right)\det\left(U^t\right) = \det\left(BB^t\right),$$

where we used $\det(UU^t) = \det(U)^2 = 1$. Hence $\sqrt{\det(B'B'^t)} = \sqrt{\det(BB^t)} = \det(\mathcal{L})$. □

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

Lemma 5.1. *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*

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

Proof. Since $\det(B^t) = \det(B)$, then

$$\det(\mathcal{L}) = \sqrt{\det(BB^t)} = \sqrt{\det(B)\det(B^t)} = \sqrt{\det(B)^2} = |\det(B)|.$$

□

The determinant of a lattice can be considered as the volume of its fundamental domain.

Definition 5.3. *Let $\mathcal{L}$ be a lattice with a basis $(b_1 \cdots, b_d)$. The fundamental domain or parallelepiped for $\mathcal{L}$ is the set*

$$\mathcal{P}(b_1 \cdots, b_d) = \left\{ \sum_{i=1}^{d} x_i b_i, \mid 0 \leq x_i < 1 \right\}.$$

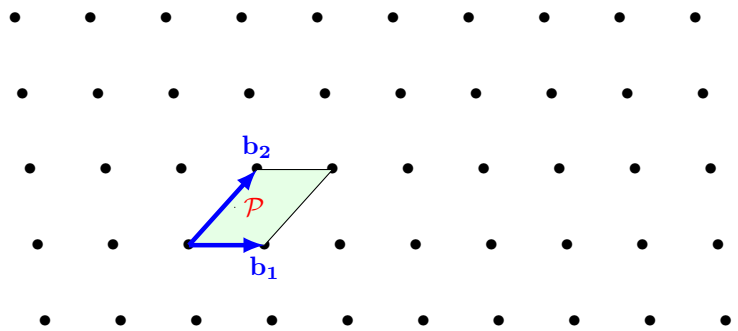


Figure 2: The fundamental domain for the basis (b_1, b_2)

Proposition 5.4. *Let $\mathcal{L}$ be a lattice with a basis $(b_1, \ldots, b_d)$. Then the volume $\mathcal{V}$ of the fundamental domain $\mathcal{P}(b_1, \ldots, b_d)$ satisfies*

$$\mathcal{V}(\mathcal{P}(b_1, \ldots, b_d)) = \det(\mathcal{L}).$$

The former result shows that any two bases of a lattice have the same volume $\mathcal{V}$ of the fundamental domain. This shows again that $\det(\mathcal{L})$ is an important invariant in a lattice.

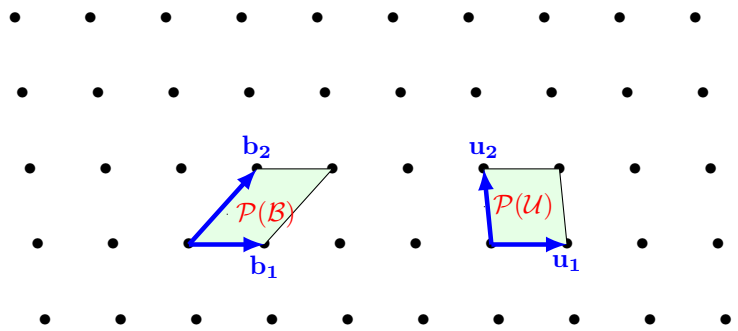


Figure 3: The fundamental domain for the bases (b_1, b_2) and (u_1, u_2)

5.3 Short vectors

Lattices are used as a fundamental tool for cryptanalysis of various public key cryptosystems such as knapsack cryptosystems, RSA, NTRU and GGH. On the other hand, lattices are used as a theoretical tool for security analysis

of several cryptosystems such as NTRU and LWE. These cryptosystems are related to hard computational problems in the theory of lattices such shortest nonzero vectors and minimal distances.

Definition 5.4. *Let $\mathcal{L}$ be a lattice. The minimal distance λ_1 of $\mathcal{L}$ is the length of the shortest nonzero vector of $\mathcal{L}$:*

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

or equivalently

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

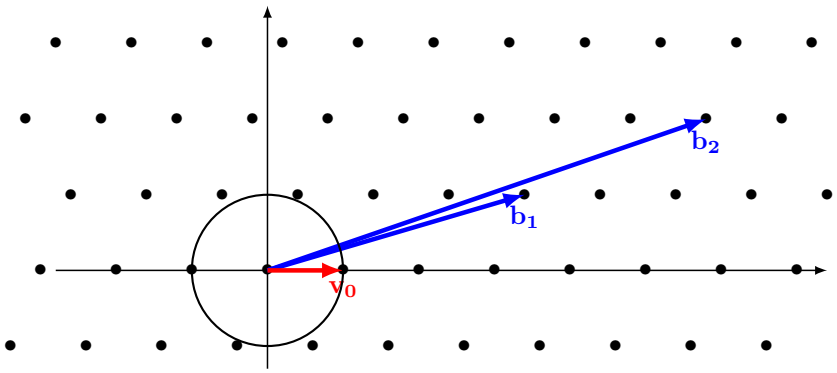


Figure 4: The shortest vectors are v_0 and $-v_0$

Definition 5.5. *Let L be a lattice of dimension n. For $i = 1, \ldots n$, the ith successive minimum of the lattice is*

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

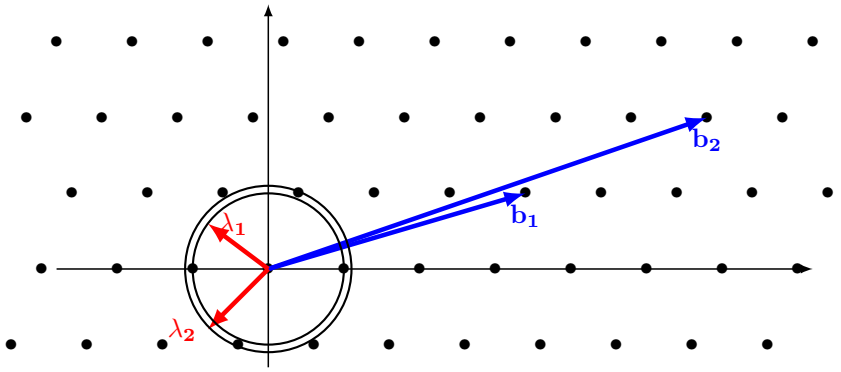


Figure 5: The first minima λ_1 and the second minima λ_2

Finding a vector v such that $\|v\| = \lambda_1$ is very hard in general. Nevertheless, in low dimension, the problem can be solved. For example, in dimension 2, the LLL algorithm (Lenstra et al., 1982) finds a basis (b_1, b_2) such that $\|b_1 = \lambda_1$ and $\|b_2\| = \lambda_2$.

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 Laarhoven et al. (2012).

Definition 5.6. *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 $\|v\| = \lambda_1(\mathcal{L})$.*

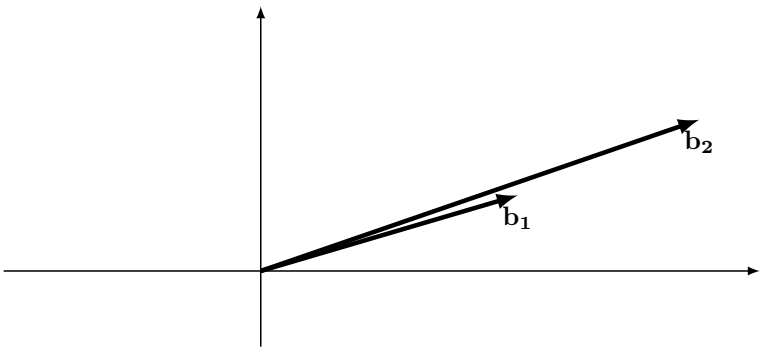


Figure 6: Find the shortest non-zero vector of the lattice with basis (b_1, b_2).

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 $\|v-u\| = d(v,\mathcal{L})$ where $d(v,\mathcal{L}) = \min_{u\in\mathcal{L}} \|v-u\|$.*

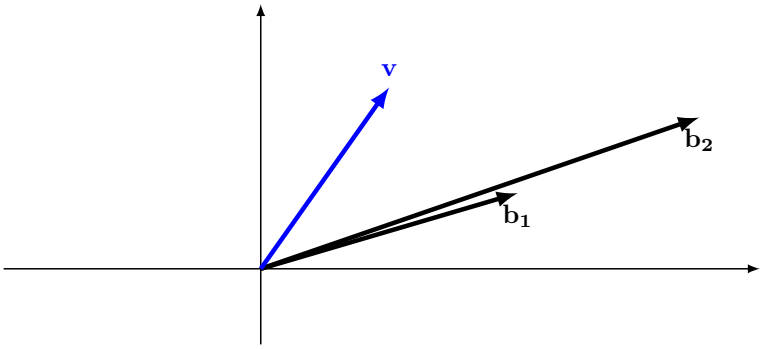


Figure 7: Find the closest vector $v_0 \in \mathcal{L}$ to $v \notin \mathcal{L}$

3. ***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 $\mathcal{L}$.*

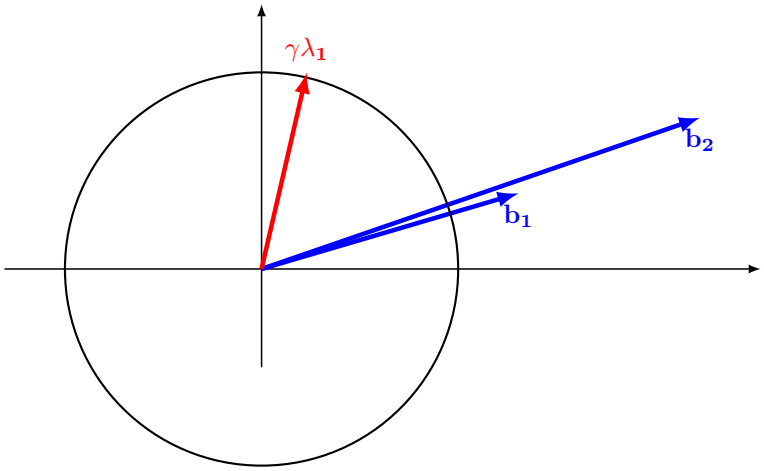


Figure 8: Find a vector v such that $\|v\| \leq \gamma\lambda_1(\mathcal{L})$

4. ***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 $\|v - u\| \leq \gamma\lambda_1 d(v, \mathcal{L})$ where $d(v, \mathcal{L}) = \min_{u\in\mathcal{L}} \|v - u\|$.*

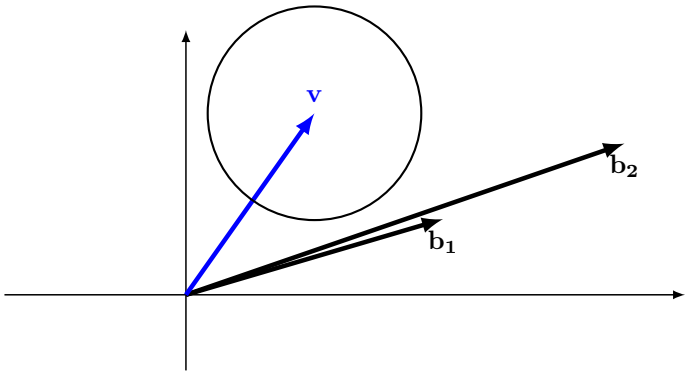


Figure 9: Find a vector $v \in \mathcal{L}$ with $v \notin \mathcal{L}$ and $\|v - u\| \leq \gamma\lambda_1 d(v, \mathcal{L})$

5. ***The Shortest Independent Vectors Problem (SIVP):*** *Given a basis matrix B for $\mathcal{L}$ of dimension n, 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 $\mathcal{L}$.

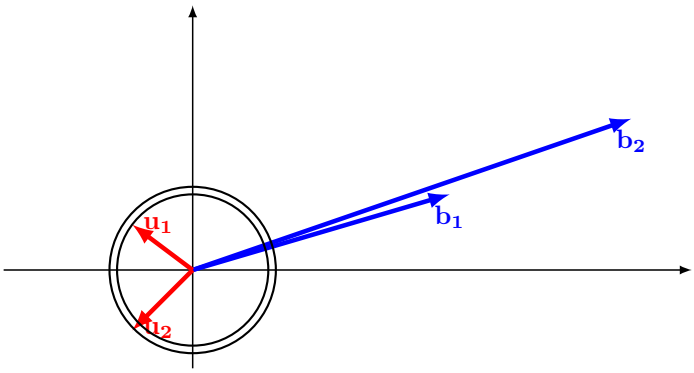


Figure 10: The two shortest independent vectors of the lattice

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 5.1 (Minkowski)**.** *Let $\mathcal{L}$ be a lattice with dimension n. Then there exists a nonzero vector $v \in \mathcal{L}$ satisfying*

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

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

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

6. Post Quantum Cryptosystems

In this section, we present two types of cryptosystems that are believed to resist to quantum computers. Both are are based on hard problems in lattices.

Lattice-based cryptography is a novel and promising type of cryptography. We review two of these developments: the NTRU cryptosystem and the LWE cryptosystem.

6.1 NTRU

NTRU was first proposed in 1996 as a very fast public key cryptosystem by Hoffstein, Pipher and Silverman. The security of NTRU is based on the hardness of some lattice problems, namely the shortest vector problem (SVP) and the closest vector problem (CVP). Both these problems have been studied extensively, and are known to be NP-hard. On the other hand, a number of connections have been established between quantum computation and SVP and CVP. Nevertheless, there are no efficient quantum algorithms for solving SVP and CVP.

NTRU (Hoffstein et al., 1998) operations take place in the quotient ring of polynomials $\mathcal{R} = \mathbb{Z}[X]/\left(X^N - 1\right)$. where N is an odd prime. Addition of two elements in $\mathcal{R}$ is defined as pairwise addition of coefficients of the same degree and multiplication is defined by the cyclic convolution product, denoted by $*$. The NTRU cryptosystem works with many parameters.

- Two relatively prime integers p and q.
- Four subsets $\mathbb{L}_f$, $\mathbb{L}_g$, $\mathbb{L}_r$, $\mathbb{L}_m$ of $\mathcal{R}$ used for key generation and encryption. The polynomials in these subsets have a few and very small coefficients.

In NTRU, the key generation, encryption and decryption primitives are as follows:

1. **Key generation**
   - Randomly choose a polynomial $f \in \mathbb{L}_f$ such that f is invertible in $\mathcal{R}$ 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 $g \in \mathbb{L}_g$.
   - Compute $h \equiv p * g * f_q \pmod{q}$.
   - Publish the public key (N, h) and the set of parameters p, q, $\mathbb{L}_f$, $\mathbb{L}_g$, $\mathbb{L}_r$ and $\mathbb{L}_m$.
   - Keep the private key (f, F_p).

2. **Encryption**
   - Represent the message as a polynomial $m \in \mathbb{L}_m$.
   - Randomly choose a polynomial $r \in \mathbb{L}_r$.
   - Encrypt m with the public key (N, h) using the rule $e \equiv r * h + m \pmod{q}$.
3. **Decryption**
   - The receiver computes $a \equiv f * e \pmod{q}$.
   - Using a centering procedure, try to recover the integer polynomial $p * r * g + f * m \pmod{q}$ from a.
   - Compute $m \equiv f_p * a \pmod{p}$.

In NTRU, the correctness of the decryption is as follows. We have

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

Then, if $p * r * g + f * m$ is an integer polynomial with coefficients in $\left[-\frac{q}{2}, \frac{q}{2}\right[$, then

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

In some situations (seeCoppersmith and Shamir (1997)), lattice attacks can be applied to recover the private key (f, F_p) in NTRU. To this end, a lattice $\mathcal{L}$ is derived from the public key (N, h) and the private key (f, g) can be recovered as likely the shortest vector in $\mathcal{L}$. Hence, if an attacker could solve the shortest vector problem (SVP), then he would be able to recover the secret key (f, g) and then (f, F_p). Nevertheless, when the NTRU parameters are properly chosen, NTRU is resistant to this attack.

6.2 LWE

In 2005, Regev (2009) invented a new cryptosystem, called learning with errors (LWE). Moreover, he found a proof of security for LWE, namely a remarkable connection between lattices and LWE: the search version of LWE is at

least as hard as quantumly approximating two problems in lattices in the worst case, GapSVP and SIVP. LWE key generation, encryption and decryption are as follows.

1. **Key generation**
   - Choose integers n, m, t, r, q and a real $\alpha > 0$.
   - Choose an error distribution χ over $\mathbb{Z}$.
   - Choose a matrix $S \in \mathbb{Z}_q^{n\times l}$ uniformly at random.
   - Choose a matrix $A \in \mathbb{Z}_q^{m\times n}$ uniformly at random.
   - Choose a matrix $E \in \mathbb{Z}_q^{m\times l}$ by choosing each entry according to a probability distribution χ on $\mathbb{Z}_q$, typically taken to be a normal distribution.
   - Compute $B = AS + E \in \mathbb{Z}_q^{m\times l}$
   - Publish the public key (A, B) and the set of parameters n, m, t, r, q and the real α.
   - Keep the private key S.
2. **Encryption**
   - Represent the message as a vector $m \in \mathbb{Z}_t^l$.
   - Choose a uniformly random vector $v_0 \in \{-r, \ldots, r\}$.
   - Encrypt m with the public key (A, B) using the rule
   $$U = A^T v_0 \in \mathbb{Z}_q^n, \quad V = B^T v_0 + f(m) \in \mathbb{Z}_q^l,$$
   where f is the function
   $$\begin{array}{rccc} f & : & \mathbb{Z}_t^l & \longrightarrow & \mathbb{Z}_q^l \\ & & (x_1, \ldots, x_l) & \longmapsto & ([x_1 q/t], \ldots, [x_l q/t]), \end{array}$$
   and $[x]$ is the nearest integer to x. The encrypted message is (U, V).
3. **Decryption**
   - Given the encrypted message (U, V), the receiver uses the private key S and computes $m = f^{-1}\left(V - S^T U\right)$, where f^{-1} is the inverse function:
   $$\begin{array}{rccc} f^{-1} & : & \mathbb{Z}_q^l & \longrightarrow & \mathbb{Z}_t^l \\ & & (y_1, \ldots, y_l) & \longmapsto & ([y_1 t/q], \ldots, [y_l t/q]). \end{array}$$

When we perform $f^{-1}\left(V - S^T U\right)$, we get

$$\begin{aligned} f^{-1}\left(V - S^T U\right) &= f^{-1}\left(B^T v_0 + f(m) - S^T\left(A^T v_0\right)\right) \\ &= f^{-1}\left((AS+E)^T v_0 + f(m) - S^T A^T v_0\right) \\ &= f^{-1}\left(E^T v_0 + f(m)\right). \end{aligned}$$

Since $v_0 \in \{-r, \ldots, r\}$ and the entries of the matrix $E \in \mathbb{Z}_q^{m\times l}$ are very small, then

$$f^{-1}\left(E^T v_0 + f(m)\right) = f^{-1}\left(f(m)\right) = m.$$

Informally, in the LWE-problem we are given a uniformly chosen matrix $A \in \mathbb{Z}_q^{m\times n}$ and a vector $B = AS + E \in \mathbb{Z}_q^{m\times l}$ where $S \in \mathbb{Z}_q^{n\times l}$ is an unknown matrix and $E \in \mathbb{Z}_q^{m\times l}$ is a vector consisting of small errors, chosen uniformly based on the normal probability distribution. The problem is then to recover the vector S. The set of parameters n, m, t, r, q and α are chosen to guarantee the security and the efficiency of the LWE cryptosystem.

7. Conclusion

Post Quantum Cryptography is a promising area of research that had emerged after the discovery of Shor's algorithm. The prominent cryptosystems like RSA and ElGamal will be totally obsolete with a quantum computer. Nevertheless, some cryptosystems, called post quantum cryptosystems, are believed to resist classical computers and quantum computers. The most known candidates belong to Hash-based cryptosystems, Code-based cryptosystems and Lattice-based cryptosystems.

References

Bennett, C. H. and Brassard, G. (1984). Quantum Cryptography: Public Key Distribution and Coin Tossing. In *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, New York. IEEE Press.

Buhler, J., Lenstra, H., and Pomerance, C. (1994). The development of the number field sieve, Volume 1554 of Lecture Notes in Computer Science.

Coppersmith, D. and Shamir, A. (1997). Lattice attacks on NTRU. In *Eurocrypt*, volume 1233, pages 52–61. Springer.

Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472.

FIPS, P. (2000). 186-2. Digital Signature Standard (DSS). *National Institute of Standards and Technology (NIST)*, 20:13.

Hoffstein, J., Pipher, J., and Silverman, J. H. (1998). NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer.

Koyama, K., Maurer, U. M., Okamoto, T., and Vanstone, S. A. (1991). New public-key schemes based on elliptic curves over the ring $\mathbb{Z}_n$. In *Annual International Cryptology Conference*, pages 252–266. Springer.

Laarhoven, T., van de Pol, J., and de Weger, B. (2012). Solving Hard Lattice Problems and the Security of Lattice-Based Cryptosystems. *IACR Cryptology EPrint Archive*, 2012:533.

Lenstra, A. K., Lenstra, H. W., and Lovász, L. (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534.

Miller, V. S. (1985). Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer.

Rabin, M. O. (1979). Digitalized signatures and public-key functions as intractable as factorization. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE.

Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332.

Smith, P. J. and Lennon, M. J. (1993). LUC: A New Public Key System. In *SEC*, pages 103–117.