UNIVERSITÉ DE LIMOGES

FACULTÉ DES SCIENCES ET TECHNIQUES

Dang Truong MAC
Fatima Zahra LABLARHI

LATTICE PROBLEMS AND APLICATIONS IN CRYPTOSYSTEMS

Supervisor: Dr. Duong Hieu PHAN

Limoges - 2017

Contents

Abstract

Lattice-based cryptography is the use of conjectured hard problems on point lattices in $\mathbb{R}^n$ as the foundation for secure cryptographic systems. Its constructions hold a great promise for post-quantum cryptography, as they enjoy very strong security proofs based on worst-case hardness, relatively efficient implementations, as well as great simplicity. In addition, lattice-based cryptography is believed to be secure against quantum computers. Our focus here will be mainly on the practical aspects of lattice-based cryptography: SIS (Short Integer Solutions) and some of its applications.

1. Motivations

Nowadays, everybody is using a secret key, from military intelligence and banking transactions to rendezvous notes. If we suppose that Alice (sender) wishes to send a private message, or plaintext, to Bob (receiver), while a nosy Eve tries to uncover the message. Alice could then encrypt the message using a secret code, so Eve will not be able to have access to it, provided of course the "encryption/decryption algorithm" is known only to Alice and Bob. Actually, and until the 1970s, all cryptosystems operate on a principle known as private key cryptography. One such protocol is the "One-Time Pad Encryption" (OTP), proposed by Vernam in 1926. The OTP is significant because it has been proven to have "perfect secrecy", by which we mean that knowledge of the key gives absolutely no additional information about the plaintext. However, there is a major disadvantage in using OTP, as each key can only be used once and has to be as long as the plaintext, one would need a large number of different keys physically distributed to both Alice and Bob. This limitation becomes even more severe in the modern context when large amounts of information have to be securely communicated, and without having Alice and Bob meeting face to face. That is why, modern cryptography employs schemes known as public key crytosystems to distribute the key. The security of these schemes relies on the fact that the key has a high computational complexity, i.e., these schemes can be broken in principle, but only with a substantial amount of computational power, which we deemed is beyond that available to Eve. For example, the RSA cryptosystem rely on the computational difficulty of factorizing any large numbers with over hundreds of digits into their integers. However, with the advent of quantum computation, such computation may not be as daunting as before. In 1949, Peter Shore showed how a quantum computer could be used to factor large numbers efficiently by making use of coherent manipulations on many quantum systems. This would certainly undermine the security of many cryptosystems currently employed. There is thus a need to search for alternative solution.

2. Introduction to lattice

2.1. Basic definitions

Definiton 2.1. *An n-dimensional lattice $\mathcal{L}$ is any subset of $\mathbb{R}^n$ that is both:*

1. *an* additive subgroup*:* $\mathbf{0} \in \mathcal{L}$*, and* $-\mathbf{x}, \mathbf{x}+\mathbf{y} \in \mathcal{L}$ *for every* $\mathbf{x}, \mathbf{y} \in \mathcal{L}$; *and*
2. discrete*: every* $\mathbf{x} \in \mathcal{L}$ *has a neighborhood in* $\mathbb{R}^n$ *in which* $\mathbf{x}$ *is the only lattice point.*

The *minimum distance* of a lattice $\mathcal{L}$ is the length of a shortest nonzero lattice vector:

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

Note that from the discreteness, and the fact that $\mathbf{0} \in \mathcal{L}$, the number $\lambda_1(\mathcal{L})$ always exists. We call $\lambda_1(\mathcal{L})$ the first successive minimum of the given lattice $\mathcal{L}$.

Generally, the ith successive minimum $\lambda_i(\mathcal{L})$ is the smallest $r > 0$ such that $\mathcal{L}$ has i linearly independent vectors of norm (or length) at most r.

Definiton 2.2 (Bases and fundamental parallelopipeds). *Every lattice $\mathcal{L}$ is always finitely generated as the integer linear combinations of some linearly independent* basis *vectors $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$:*

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \mathbf{B} \cdot \mathbb{Z}^k = \left\{ \sum_{i=1}^{k} z_i \mathbf{b}_i \middle| z_i \in \mathbb{Z} \right\}.$$

The k is called that rank *of the basis, and is an invariant of the lattice. When $k = n$, the lattice is called* full-rank*. From now on, we restrict ourselves to full-rank lattices. For a lattice $\mathcal{L}$ having basis $\mathbf{B}$, the set*

$$\mathcal{P}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \middle| 0 \leq x_i < 1 \text{ for all } 1 \leq i \leq n \right\}$$

is called the fundamental parallelopiped *of $\mathcal{L}$.*

Remark 2.1. *From the definition we see that $\mathcal{P}(\mathbf{B})$ contains only one lattice point, which is the origin, and every translate of it also contains exactly one lattice point.*

Definiton 2.3 (The dual lattice). *The* dual *of a lattice $\mathcal{L} \subset \mathbb{R}^n$ is defined as*

$$\mathcal{L}^* := \{\mathbf{w} \in \mathbb{R}^n \big| \langle \mathbf{w}, \mathcal{L} \rangle \subseteq \mathbb{Z}\},$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors.

Assume we know a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ of $\mathcal{L}$, and define the vectors $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*$ such that $\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle = \delta_{ij}$, where δ_{ij} is the Kronecker symbol. Then one can easily verify that $\mathcal{L}^* = \mathbf{B}^* \cdot \mathbb{Z}^n$, where $\mathbf{B}^*$ is the matrix whose columns are the vectors $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*$. (This is the reason why sometimes we call *reciprocal* lattice.)

Remark 2.2. *The concept of dual lattice plays an important role in the geometry of numbers theory as well as in lattice cryptography.*

2.2. Computational Problems

Problem 1 (Shortest Vector Problem (SVP)). *Given an arbitrary basis $\mathbf{B}$ of some lattice $\mathcal{L}$, find a shortest nonzero lattice vector, i.e., a $\mathbf{v} \in \mathcal{L}$ for which $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$.*

Problem 2 (Approximate Shortest Vector Problem SVP$_\gamma$)). *Given a basis $\mathbf{B}$ of an n-dimensional lattice $\mathcal{L}$, find a nonzero vector $\mathbf{v} \in \mathcal{L}$ for which $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$.*

Problem 3 (**Decision Approximate SVP (GapSVP$_\gamma$)).** *Given a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\mathcal{L}$ *where either* $\lambda_1(\mathcal{L}) \leq 1$ *or* $\lambda_1(\mathcal{L}) > \gamma(n)$*, determine which is the case.*

Problem 4 (**Approximate Short Independent Vectors Problem (SIVP$_\gamma$)).** *Given a basis* $\mathbf{B}$ *of a full-rank* n*-dimensional lattice* $\mathcal{L}$*, output a set* $\mathbf{S} = \{\mathbf{s}_i\} \subset \mathcal{L}$ *of* n *linearly independent lattice vectors where* $\|\mathbf{s}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$*.*

Problem 5 (**Bounded Distance Decoding Problem (BDD$_\gamma$)).** *Given a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\mathcal{L}$ *and a target point* $\mathbf{t} \in \mathbb{R}^n$ *with the guarantee that* $\mathrm{dist}(\mathbf{t}, \mathcal{L}) < d$*, where* $d = \frac{\lambda_1(\mathcal{L})}{2\gamma(n)}$*, find the unique lattice vector* $\mathbf{t} \in \mathcal{L}$ *such that* $\|\mathbf{t} - \mathbf{v}\| < d$*.*

Remark 2.3. *These problems are believed to be "hard" to solve.*

2.3. Gaussian Distributions

Definiton 2.4 (**Gaussians).** *For any positive integer* $n, \mathbf{c} \in \mathbb{R}^n$ *and real* $s > 0$*, which is taken to be* $s = 1$ *when omitted, define the* Gaussian function $\rho_s : \mathbb{R}^n \to \mathbb{R}^+$ *of* parameter *(or* width*)* s *centered in* $\mathbf{c}$ *as*

$$\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp\left(-\frac{\pi\|\mathbf{x} - \mathbf{c}\|^2}{s^2}\right).$$

The total measure associated to $\rho_{s,\mathbf{c}}$ is

$$\int_{\mathbb{R}^n} \rho_{s,\mathbf{c}}(\mathbf{x}) d\mathbf{x} = s^n.$$

Indeed, to see this we just need to consider the case $\mathbf{c} = \mathbf{0}$ and $s = 1$. Then we have

$$\begin{aligned} \int_{\mathbb{R}^n} \rho(\mathbf{x}) d\mathbf{x} &= \int_{\mathbb{R}} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} e^{-\pi(\sum_{i=1}^n x_i^2)} dx_1 dx_2 \cdots dx_n \\ &= \left(\int_{\mathbb{R}} e^{-\pi x^2} dx\right)^n \\ &= 1. \end{aligned}$$

Therefore, we can define the continuous Gaussian distribution around $\mathbf{c}$ with parameter s by its probability density function

$$\forall \mathbf{x} \in \mathbb{R}^n, D_{s,\mathbf{c}(\mathbf{x})} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{s^n}.$$

It can be seen that the expected square distance from $\mathbf{c}$ of a vector chosen from this distribution is $\frac{ns^2}{2\pi}$. Indeed, as above we consider the case $s = 1$ and $\mathbf{c} = 0$

$$\begin{aligned} \mathbb{E}_{\mathbf{x} \sim D_{s,\mathbf{c}}}[\|\mathbf{x}\|^2] &= \int_{\mathbb{R}^n} \mathbf{x}^2 \rho(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbb{R}^n} \left(\sum_{i=1}^n x_i^2\right) e^{-\pi(\sum_{i=1}^n x_i^2)} dx_1 \cdots dx_n \\ &= \frac{n}{2\pi}, \end{aligned}$$

by using induction on n. So, intuitively, one can think of $D_{s,\mathbf{c}}$ as a sphere of radius $s\sqrt{\frac{n}{2\pi}}$ For a countable set A, we define $\rho(A) = \sum_{\mathbf{x}\in A} \rho(\mathbf{x})$. For any vector $\mathbf{c}$, real $s > 0$, and lattice $\mathcal{L}$, we define the probability distribution $D_{\Lambda,s,\mathbf{c}}$ over Λ by

$$\forall \mathbf{x} \in \Lambda, D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{D_{s,\mathbf{c}}(\mathbf{x})}{D_{s,\mathbf{c}}(\Lambda)} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}.$$

We will use the connection between $D_{s,\mathbf{c}}$ and $D_{\Lambda,s,\mathbf{c}}$: if $\mathbf{x}$ is distributed according to $D_{s,c}$ and we condition on $x \in \Lambda$, then the conditional distribution of $\mathbf{x}$ is $D_{\Lambda,s,\mathbf{c}}$. With some special condition, $D_{\Lambda,s,\mathbf{c}}$ behaves in many respects like the continuous Gaussian distribution.

To measure the distance between two random variables, we define statistical distance and state some useful facts that will be used.

Definiton 2.5. *We define the statistical distance between two discrete random variables X and Y over a (countable) set A as*

$$\Delta(X, Y) = \frac{1}{2} \sum_{a\in A} \big|\mathbb{P}[X = a] - \mathbb{P}[Y = a]\big|.$$

Similarly, for two continuous random variables X and Y over $\mathbb{R}^n$ with probability density functions T_1 and T_2 respectively, the statistical distance is defined as

$$\Delta(X, Y) = \frac{1}{2} \int_{\mathbb{R}^n} \big|T_1(\mathbf{x}) - T_2(\mathbf{x})\big| d\mathbf{x}.$$

Proposition 2.1. *Let $X, Y, X_1, \ldots, X_k, Y_1, \ldots, Y_k$ be totally independent random variables, and f a random function. Then*

(i) $\Delta(f(X), f(Y)) \leq \Delta(X, Y)$.

(ii) $\Delta((X_1, \ldots, X_k), (Y_1, \ldots, Y_k)) \leq \sum_{i=1}^{k} \Delta(X_i, Y_i)$.

Definiton 2.6 (Fourier transform). *The Fourier transform of a function $h : \mathbb{R}^n \to \mathbb{R}$ is defined to be*

$$\hat{h}(\mathbf{w}) = \int_{\mathbb{R}^n} h(\mathbf{x}) e^{-2\pi i\langle \mathbf{x},\mathbf{w}\rangle} d\mathbf{x}.$$

If h is defined by $h(\mathbf{x}) = g(\mathbf{x} + \mathbf{v})$ for some function g and vector $\mathbf{v}$ then we have

$$\hat{h}(\mathbf{x}) = e^{2\pi i\langle \mathbf{v},\mathbf{w}\rangle} \hat{g}(\mathbf{x}). \tag{1}$$

Indeed, by the definition we have

$$\begin{aligned}\hat{h}(\mathbf{w}) &= \int_{\mathbb{R}^n} h(\mathbf{x}) e^{-2\pi i\langle \mathbf{x},\mathbf{w}\rangle} d\mathbf{x} \\ &= \int_{\mathbb{R}^n} g(\mathbf{x} + \mathbf{v}) e^{-2\pi i\langle \mathbf{x}+\mathbf{v}-\mathbf{v},\mathbf{w}\rangle} d(\mathbf{x} + \mathbf{v}) \\ &= e^{2\pi i\langle \mathbf{v},\mathbf{w}\rangle} \int_{\mathbb{R}^n} g(\mathbf{x} + \mathbf{v}) e^{-2\pi i\langle \mathbf{x}+\mathbf{v},\mathbf{w}\rangle} d(\mathbf{x} + \mathbf{v}) \\ &= e^{2\pi i\langle \mathbf{v},\mathbf{w}\rangle} \hat{g}(\mathbf{w}).\end{aligned}$$

Similarly, if h is defined as $h(\mathbf{x}) = e^{2\pi i\langle \mathbf{x},\mathbf{v}\rangle} g(\mathbf{x})$ for some function g and vector $\mathbf{v}$ then

$$\hat{h}(\mathbf{w}) = \hat{g}(\mathbf{w} - \mathbf{v}).$$

If we denote by $h^{\mathbf{u}}$ the derivative of h in the direction of some unit vector $\mathbf{u}$ then its Fourier transform is

$$\widehat{h^{\mathbf{u}}}(\mathbf{w}) = 2\pi i\langle \mathbf{u}, \mathbf{w}\rangle \cdot \hat{h}(\mathbf{w}).$$

An important fact is that the Gaussian is its own Fourier transform, i.e., $\hat{\rho} = \rho$ and $\hat{\rho}_s = s^n \rho_{1/s}$. We have the following lemma

Lemma 2.1. *For any lattice Λ,*

$$\hat{\rho}(\Lambda) = \det(\Lambda^*) \cdot \hat{\rho}(\Lambda^*).$$

Proof. This follows directly from Poisson Summation Formula. □

Lemma 2.2 (Banaszczyk)**.** *For any $c > (2\pi)^{-1/2}$, n-dimensional lattice Λ, and vector $\mathbf{v} \in \mathbb{R}^n$,*

$$\rho(\Lambda \setminus c\sqrt{n}\mathcal{B}) < C^n \cdot \rho(\Lambda),$$
$$\rho((\Lambda + \mathbf{v}) \setminus c\sqrt{n}\mathcal{B}) < 2C^n \cdot \rho(\Lambda),$$

where $C = c\sqrt{2\pi e} \cdot e^{-\pi c^2}$.

Proof. See [4]. □

2.4. Smoothing parameter

Definiton 2.7. *For an n-dimensional lattice Λ, and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^*) \leq 1 + \epsilon$,*

$$\eta_\epsilon(\Lambda) = \min\{s > 0 \big| \rho_{1/s}(\Lambda^*) \leq 1 + \epsilon\}.$$

Note that $\eta_\epsilon(\Lambda)$ is well-defined because $\rho_{1/s}(\Lambda^*)$ is a continuous and strictly decreasing function of s with $\lim\limits_{s\to 0} \rho_{1/s}(\Lambda^*) = \infty$ and $\lim\limits_{s\to\infty} \rho_{1/s}(\Lambda^*) = 1$.

Theorem 2.1 (Micciancio)**.** *For any full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$, we have*

$$\eta_{2^{-n}}(\mathcal{L}) \leq \frac{\sqrt{n}}{\lambda_1(\mathcal{L}^*)}.$$

Proof. In Lemma 2.2, let $c = 1$ then $C = e^{-\pi^2}\sqrt{2\pi e}$. We have

$$\rho(\Lambda \setminus \sqrt{n}\mathcal{B}) < \frac{C^n}{1 - C^n} \rho(\Lambda \cap \sqrt{n}\mathcal{B}).$$

If we choose s such that $s > \sqrt{n}/\lambda_1(\Lambda^*)$ then a shortest vector in $s\Lambda^*$ is longer than $\sqrt{n}$, and therefore $s\Lambda^* \setminus \sqrt{n}\mathcal{B} = s\Lambda^* \setminus \{\mathbf{0}\}$ and $s\Lambda^* \cap \sqrt{n}\mathcal{B} = \{\mathbf{0}\}$. We have,

$$\begin{aligned}\rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) &= \rho(s\Lambda^* \setminus \sqrt{n}\mathcal{B}) \\ &< \frac{C^n}{1 - C^n} \rho(s\Lambda^* \cap \sqrt{n}\mathcal{B}) \\ &= \frac{C^n}{1 - C^n}\end{aligned}$$

It is easy to see that $C < \frac{1}{4}$ and thus $\frac{C^n}{1-C^n} < 2^{-n}$. Hence, by the definition of $\eta_{2^{-n}}(\Lambda)$ we obtain

$$\eta_{2^{-n}}(\Lambda) \leq \frac{\sqrt{n}}{\lambda_1(\Lambda^*)}.$$

□

Theorem 2.2 (Micciancio)**.** *For any n-dimensional lattice $\mathcal{L}$ and positive $\epsilon > 0$,*

$$\eta_\epsilon(\mathcal{L}) \leq \left(\frac{\log(2n(1+\epsilon^{-1}))}{\pi}\right)^{1/2} \cdot \lambda_n(\mathcal{L}).$$

In particular, for any superlogarithmic function $\omega(\log n)$, there exists a negligible function $\epsilon(n)$ such that $\eta_\epsilon(\mathcal{L}) \leq \sqrt{\omega(\log n)} \cdot \lambda_n(\mathcal{L})$.

Proof. See [3] □

The meaning of smoothing parameter is expressed in the following lemma:

Lemma 2.3. *For any $s > 0, \mathbf{c} \in \mathbb{R}^n$, and any lattice $\mathcal{L}(\mathbf{B})$, the statistical distance between $D_{s,\mathbf{c}} \mod \mathcal{P}(\mathbf{B})$ and the uniform distribution over $\mathcal{P}(\mathbf{B})$ is at most $\frac{1}{2}\rho_{1/s}(\mathcal{L}(\mathbf{B})^* \setminus \{\mathbf{0}\})$. In particular, for any $\epsilon > 0$ and any $s \geq \eta_\epsilon(\mathbf{B})$, the statistical distance is at most*

$$\Delta(D_{s,\mathbf{c}} \mod \mathcal{P}(\mathbf{B}), U(\mathcal{P}(\mathbf{B}))) \leq \frac{\epsilon}{2}.$$

Proof. Let Y be the density function of the distribution over $\mathcal{P}(\mathbf{B})$ defined by $D_{s,\mathbf{c}}$ mod $\mathcal{P}(\mathbf{B})$

$$\begin{aligned} Y(\mathbf{x}) &= \frac{1}{s^n} \sum_{\mathbf{y}\in\mathcal{L}(\mathbf{B})} \rho_{s,\mathbf{c}}(\mathbf{x}+\mathbf{y}) \\ &= \frac{1}{s^n}\rho_{s,\mathbf{c}-\mathbf{x}}(\mathcal{L}(\mathbf{B})). \end{aligned}$$

By Equation (1), the Fourier transform of $\rho_{s,\mathbf{c}-\mathbf{x}}$ at point $\mathbf{w}$ is $e^{2\pi i\langle \mathbf{x}-\mathbf{c},\mathbf{w}\rangle} s^n \rho_{1/s}(\mathbf{w})$. By Lemma 2.1,

$$Y(\mathbf{x}) = \det(\mathcal{L}(\mathbf{B})^*) \sum_{\mathbf{w}\in\mathcal{L}(\mathbf{B})^*} e^{2\pi i\langle \mathbf{x}-\mathbf{c},\mathbf{w}\rangle} \rho_{1/s}(\mathbf{w}).$$

The density function of the uniform distribution over $\mathcal{P}(\mathbf{B})$ is $U(\mathbf{x}) = \frac{1}{\text{vol}(\mathcal{P}(\mathbf{B}))} = \det(\mathcal{L}(\mathbf{B})^*)$. Thus the statistical distance between Y and U is

$$\begin{aligned} \Delta(Y,U) &= \frac{1}{2}\int_{\mathcal{P}(\mathbf{B})} |Y(\mathbf{x}) - U(\mathbf{x})| d\mathbf{x} \\ &\leq \frac{1}{2}\text{vol}(\mathcal{P}(\mathbf{B})) \cdot \max_{\mathcal{P}(\mathbf{B})} |Y(\mathbf{x}) - \det(\mathcal{L}(\mathbf{B}))^*| \\ &= \frac{1}{2}\text{vol}(\mathcal{P}(\mathbf{B})) \cdot \det(\mathcal{L}(\mathbf{B})^*) \cdot \max_{\mathcal{P}(\mathbf{B})} \Big| \sum_{\mathbf{w}\in\mathcal{L}(\mathbf{B})^*\setminus\{\mathbf{0}\}} e^{2\pi i\langle \mathbf{x}-\mathbf{c},\mathbf{w}\rangle}\rho_{1/s}(\mathbf{w})\Big| \\ &\leq \frac{1}{2}\rho_{1/s}(\mathcal{L}(\mathbf{B})^* \setminus \{\mathbf{0}\}), \end{aligned}$$

where the last inequality comes from the triangle inequality.
The result of particular case follows from the definition of $\eta_\epsilon(\mathbf{B})$. □

To sum up the meaning of *smoothing parameter* s (with respect to a given ϵ), one can think of it as the radius of a sphere S in $\mathbb{R}^n$. Intuitively, if s is large enough then S will contain a big number of lattice points. After projecting these points into the fundamental parallelepiped, we obtain a (nearly) uniformly distribution in $\mathcal{P}(\mathbf{B})$.

3. Modern Foundations

3.1. Short Integer Solutions Problem

Definiton 3.1 (Short Integer Solution ($\mathrm{SIS}_{n,q,\beta,m}$))). *Given m uniformly random vectors $\boldsymbol{a}_i \in \mathbb{Z}_q^n$, forming the columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ of norm $\|\mathbf{z}\| \leq \beta$ such that*

$$f_{\mathbf{A}}(\mathbf{z}) := \mathbf{A}\mathbf{z} = \sum_{i=1}^{m} \mathbf{a}_i \cdot z_i = \mathbf{0} \in \mathbb{Z}_q^n.$$

The SIS problem can be seen as an average-case short vector problem on a certain family of so-called "q-ary" m-dimensional integer lattices, namely, the lattices

$$\Lambda_q(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \big| \mathbf{A}\mathbf{z} = \mathbf{0} \in \mathbb{Z}_q^n\}.$$

Remark 3.1. *1. The condition "short" is essential since otherwise we can simply choose $\mathbf{z} = (q, 0, \dots, 0) \in \mathbb{Z}^m$. This vector is obviously satisfied the equality.*

2. The norm bound β and the number m of vectors $\mathbf{a}_i$ must be large enough that a solution is guaranteed to exist. This is the case whenever $\beta \geq \sqrt{\lceil n \log q \rceil}$ and $m \geq \lceil n \log q \rceil$.

3.1.1. Hardness

In this subsection, we show that the SIS problem is hard in the sense that if there is an algorithm that can can solve the SIS problem in polynomial time then there is also a polynomial time algorithm that solve one of lattice problems. (In fact, these problems are considered to be "hard".) In particular, we consider the following lattice problem

Definiton 3.2 (Incremental Guaranteed Distance Decoding). *An input to $\mathrm{IncGDD}_{\gamma,g}^{\phi}$ is an n-dimensional lattice basis $\mathbf{B}$, a set of n linearly independent vectors $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$, a target vector $\mathbf{t}$, a positive integer g, and a real $r > \gamma(n) \cdot \phi(\mathbf{B})$. The goal is to output a lattice vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{s} - \mathbf{t}\| \leq \frac{\|\mathbf{S}\|}{g} + r$*

Remark 3.2. *1. If $\|\mathbf{B}\|$ is too small, i.e., we already have a "short" basis, then the problem can be solved by using nearest plane algorithm of Babai.*

2 We can think of $\mathbf{S}$ as a basis of a sublattice of $\mathcal{L}(\mathbf{B})$.

3. The role of r is to guarantee that such a vector $\mathbf{s}$ always exists (with respect to our setting up parameters).

4. *One can think of* InᴄGDD *as asking to find a lattice point within distance roughly $\frac{\|\mathbf{S}\|}{g}$ from the target, provided that $\|\mathbf{S}\|$ is not too small.*

Now we show how to "reduce" this problem to the SIS problem, i.e., instead of solving InᴄGDD we will solve an SIS problem whose instances are constructed based on the inputs of the above problem. Since the input to SIS are random elements in $\mathbb{Z}_q^n$, so the main point is to sample uniformly at random the group $\mathbb{Z}_q^n$. The basic ideas are followings:

1. Starting from a Gaussian distribution $D_{s,\mathbf{t}}$, where $s \geq \eta_\epsilon(\mathbf{B}), \mathbf{t} \in \mathbb{R}^n$.

2. Using the Sampling procedure (which is described in the Lemma below) to generate m pairs $(\mathbf{c}_1, \mathbf{y}_1), (\mathbf{c}_2, \mathbf{y}_2), \ldots, (\mathbf{c}_m, \mathbf{y}_m)$, such that $\mathbf{c}_i$ is distributed uniformly in $\mathcal{P}(\mathbf{B})$ and $\mathbf{y}_i \in \mathcal{L}(\mathbf{B})$ is a lattice vector closed to $\mathbf{c}_i$. (These vectors $\mathbf{c}_i$ are related to the target vector $\mathbf{t}$ of the input of InᴄGDD.)

3. Partitioning the parallelepiped $\mathcal{P}(\mathbf{B})$ into q^n smaller parallelepipeds, corresponding to elements of $\mathbb{Z}_q^n$.

4. For each $\mathbf{c}_i$, let $\bar{\mathbf{c}}_i = \mathbf{B}\lfloor q \cdot \mathbf{B}^{-1}\mathbf{c}_i\rfloor / q$. Notice that

$$\begin{aligned}\|\mathbf{c}_i - \bar{\mathbf{c}}_i\| &= \left\|\mathbf{c}_i - \frac{\mathbf{B}\lfloor q \cdot \mathbf{B}^{-1}\mathbf{c}_i\rfloor}{q}\right\| \\ &\leq \frac{\|\mathbf{B}\|}{q} \cdot \left\|q \cdot \mathbf{B}^{-1}\mathbf{c}_i - \lfloor q \cdot \mathbf{B}^{-1}\mathbf{c}_i\rfloor\right\| \\ &\leq \frac{n\|\mathbf{B}\|}{q}.\end{aligned}$$

5. Let $\mathbf{a}_i = \lfloor q \cdot \mathbf{B}^{-1}\mathbf{c}_i\rfloor \mod q \in \mathbb{Z}_q^n$ be the group element corresponding to the parallelepiped that contains $\mathbf{c}_i$. Each $\mathbf{c}_i$ is uniformly distributed in $\mathcal{P}(\mathbf{B})$ so each $\mathbf{a}_i$ is uniformly distributed in $\mathbb{Z}_q^n$.

6. Applying the procedure $\mathcal{F}$ (which is described in the Lemma below) to the matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m]$ to find a "short" vector $\mathbf{z}$ such that $\mathbf{Az} = 0$ and $\|\mathbf{z}\| \leq \beta$.

7. Let $\bar{\mathbf{C}} = [\bar{\mathbf{c}}_1, \ldots, \bar{\mathbf{c}}_m]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$. Since $\mathbf{Cz}$ is closed to both $\bar{\mathbf{C}}\mathbf{z}$ and $\mathbf{Yz}$ and that $\bar{\mathbf{C}}\mathbf{z}, \mathbf{Yz} \in \mathcal{L}(\mathbf{B})$ we obtain $(\bar{\mathbf{C}} - \mathbf{Y})\mathbf{z}$ is a lattice vector close to $\mathbf{0}$, which is a solution to SIS.

Lemma 3.1 (**Sampling Lemma**). *There is a probabilistic polynomial time algorithm $\mathcal{S}(\mathbf{B}, \mathbf{t}, s)$ that on input an n-dimensional lattice $\mathcal{L}(\mathbf{B})$, where $\mathbf{B} \in \mathbb{R}^{n \times n}$, a vector $\mathbf{t} \in \mathbb{R}^n$, and a real $s \geq \eta_\epsilon(\mathbf{B})$ (for some $\epsilon > 0$), outputs a pair of vectors $(\mathbf{c}, \mathbf{y}) \in \mathcal{P}(\mathbf{B}) \times \mathcal{L}(\mathbf{B})$ such that*

1. *the distribution of vector $\mathbf{c}$ is within statistical distance $\Delta(\mathbf{c}, U(\mathcal{P}(\mathbf{B}))) \leq \epsilon/2$ from the uniform distribution over $\mathcal{P}(\mathbf{B})$;*

2. *for any $\hat{\mathbf{c}} \in \mathcal{P}(\mathbf{B})$, the conditional distribution of $\mathbf{y}$ given $\mathbf{c} = \hat{\mathbf{c}}$ is $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}+\hat{\mathbf{c}}}$.*

Proof. The sampling procedure $\mathcal{S}(\mathbf{B}, \mathbf{t}, s)$ is the following:

1. Generate a noise vector r with probability density $D_{s,\mathbf{t}}$.
2. Output $\mathbf{c} = -\mathbf{r} \mod \mathcal{P}(\mathbf{B})$ and $\mathbf{y} = \mathbf{r} + \mathbf{c}$.

Now, notice that by Lemma 2.3 and $s \geq \eta_\epsilon(\mathbf{B})$, we have

$$\begin{aligned}\Delta(\mathbf{c}, U(\mathcal{P}(\mathbf{B}))) &= \Delta(-D_{s,\mathbf{t}} \mod \mathcal{P}(\mathbf{B}), U(\mathcal{P}(\mathbf{B}))) \\ &= \Delta(D_{s,-\mathbf{t}} \mod \mathcal{P}(\mathbf{B}), U(\mathcal{P}(\mathbf{B}))) \\ &\leq \frac{\epsilon}{2}.\end{aligned}$$

For the second property, fix any $\hat{\mathbf{c}} \in \mathcal{P}(\mathbf{B})$. Then by definition, the distribution of $\mathbf{r} + \hat{\mathbf{c}}$ is $D_{s,\mathbf{t}+\hat{\mathbf{c}}}$. Conditioning on $\mathbf{c} = \hat{\mathbf{c}}$ is the same as conditioning on $\mathbf{r} + \hat{\mathbf{c}} \in \mathcal{L}(\mathbf{B})$. The distribution of $\mathbf{r} + \hat{\mathbf{c}}$ conditioned on $\mathbf{r} + \hat{\mathbf{c}} \in \mathcal{L}(\mathbf{B})$ is $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}+\hat{\mathbf{c}}}$. □

Lemma 3.2 (**Combining Procedure**)**.** *There is a probabilistic polynomial time oracle algorithm $\mathcal{A}^{\mathcal{F}}(\mathbf{B}, \mathbf{S}, \mathbf{C}, q)$ that on input an n-dimensional lattice $\mathbf{B} \in \mathbb{R}^n$, a full-rank sublattice $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$, m vectors $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m] \in \mathcal{P}(\mathbf{B})^m$, and a positive q, makes a single oracle call $\mathcal{F}(\mathbf{A}) = \mathbf{z}$, with $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and outputs a vector $\mathbf{x} \in \mathbb{R}^n$ such that*

1. *if the input matrix $\mathbf{C}$ is distributed uniformly at random, then the query matrix $\mathbf{A}$ is also uniformly distributed;*
2. *if the oracle's answer $\mathbf{z} \in \Lambda_q(\mathbf{A})$, then the output vector $x \in \mathcal{L}(\mathbf{B})$;*
3. *the distance between $\mathbf{x}$ and $\mathbf{Cz}$ is at most $n\sqrt{m}\|\mathbf{S}\| \cdot \|\mathbf{z}\|/q$.*

Proof. The procedure $\mathcal{A}^{\mathcal{F}}(\mathbf{B}, \mathbf{S}, \mathbf{C}, q)$ is the following:

1. Generate m uniformly random lattice vectors $\mathbf{v}_i \in \mathcal{L}(\mathbf{B}) \mod \mathcal{P}(\mathbf{S})$.
2. Define the matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m]$, where $\mathbf{w}_i \equiv \mathbf{v}_i + \mathbf{c}_i \mod \mathcal{P}(\mathbf{S})$ for all $i = 1, 2, \ldots, m$.
3. Define the query $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$, where
$$\mathbf{a}_i = \lfloor q \cdot \mathbf{S}^{-1}\mathbf{w}_i \rfloor \in \mathbb{Z}_q^n \qquad \text{for all } i = 1, 2, \ldots, m.$$
4. Call the oracle $\mathcal{F}$ on input $\mathbf{A}$ to obtain an integer vector $\mathbf{z} = \mathcal{F}(\mathbf{A})$.
5. Output the vector $\mathbf{x} = (\mathbf{C} - \mathbf{W} + \mathbf{SA}/q)\mathbf{z}$.

We prove the first property. Note that if $\mathbf{c}$ is uniformly distributed in $\mathcal{P}(\mathbf{B})$ and $\mathbf{v}$ is chosen uniformly from the vectors in $\mathcal{L}(\mathbf{B}) \mod \mathcal{P}(\mathbf{S})$, then $\mathbf{c} + \mathbf{v} \mod \mathcal{P}(\mathbf{S})$ is distributed uniformly in $\mathcal{P}(\mathbf{S})$. This holds since the sets $(\mathbf{v} + \mathcal{P}(\mathbf{B})) \mod \mathcal{P}(\mathbf{S})$ for all $v \in \mathcal{L}(\mathbf{B}) \mod \mathcal{P}(\mathbf{S})$ form a partition of $\mathcal{P}(\mathbf{S})$ into sets of equal volume. Thus we see that if $\mathbf{C} \in \mathcal{P}(\mathbf{B})^m$ is distributed uniformly then $\mathbf{W}$ is distributed uniformly in $\mathcal{P}(\mathbf{B})^m$. From this it is easily follows that $\mathbf{A}$ is distributed uniformly in $\mathbb{Z}_q^{n\times m}$.

For the second property, assume that $\mathbf{z} \in \Lambda_q(\mathbf{A})$, and consider the output vector

$$\begin{aligned}\mathbf{x} &= (\mathbf{C} - \mathbf{W} + \mathbf{SA}/q)\mathbf{z} \\ &= \sum_{i=1}^{m}(\mathbf{c}_i - \mathbf{w}_i)z_i + \mathbf{S}(\mathbf{Az}/q.\end{aligned}$$

Notice that for $i = 1, 2, \ldots, m$ the vector

$$\mathbf{c}_i - \mathbf{w}_i = \mathbf{c}_i + \mathbf{v}_i - \mathbf{w}_i - \mathbf{v}_i$$

belongs to $\mathcal{L}(\mathbf{B})$ because $\mathbf{w}_i \equiv \mathbf{v}_i + \mathbf{c}_i$ modulo $\mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B})$ and $\mathbf{v}_i \in \mathcal{L}(\mathbf{B})$. Also $\mathbf{Az}/q$ is an integer vector because $\mathbf{Az} = \mathbf{0} \mod q$. This proves that $\mathbf{x} \in \mathcal{L}(\mathbf{B})$.

For the third property, we have

$$\begin{aligned}\|\mathbf{x} - \mathbf{Cz}\| &= \left\|\sum_{i=1}^{m}(\mathbf{w}_i - (\mathbf{S}/q)\mathbf{a}_i)z_i\right\| \\ &= \frac{1}{q}\left\|\mathbf{S}\sum_{i=1}^{m}(\mathbf{u}_i - \lfloor\mathbf{u}_i\rceil)z_i\right\|,\end{aligned}$$

where $\mathbf{u}_i = q\mathbf{S}^{-1}\mathbf{w}_i$. Since for each i, all coordinates of $\mathbf{u}_i - \lfloor\mathbf{u}_i\rceil$ are bounded by 1, the vector $\sum_{i=1}^{m}(\mathbf{u}_i - \lfloor\mathbf{u}_i\rceil)z_i$ has all coordinates bounded by $\sum_i |z_i| \leq \|\mathbf{z}\|\sqrt{m}$. It follows by triangle inequality that

$$\left\|\mathbf{S}\sum_{i=1}^{m}(\mathbf{u}_i - \lfloor\mathbf{u}_i\rceil)z_i\right\| \leq n\sqrt{m}\|\mathbf{z}\|\|\mathbf{S}\|,$$

and $\|\mathbf{x} - \mathbf{Cz}\| \leq n\sqrt{m}\|\mathbf{z}\|\|\mathbf{S}\|/q$. □

We have the following theorem:

Theorem 3.1. *For any function $g(n) > 0$, polynomially bounded functions $m(n)$ and $\beta(n) = n^{O(1)}$, negligible function $\epsilon(n) = n^{-\omega(1)}$, and $q(n) = g(n)\beta(n) \cdot n\sqrt{m(n)}$, there is a probabilistic polynomial time reduction from solving $\text{IncGDD}^{\eta_\epsilon}_{\gamma,g}$ for $\gamma(n) = \beta(n)\sqrt{n}$ on $n-$dimensional instances in the worst case to solving $\text{SIS}_{q,m,\beta}$ on the average with non-negligible probability.*

To sum up the *main ideas* of the reduction, starting from the input $\mathbf{t} \in \mathbb{R}^n$ of the IncGDD we construct m vectors $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_m$, and give these vectors as inputs of the Sample procedure to obtain m pairs $(\mathbf{c}_i, \mathbf{y}_i) \in \mathcal{P}(\mathbf{B}) \times \mathcal{L}(\mathbf{B}), 1 \leq i \leq m$. The $\mathbf{c}_i$ is (nearly) uniformly distributed in $\mathcal{P}(\mathbf{B})$ and $\mathbf{y}_i$ is distributed according to a discrete Gaussian distribution over $\mathcal{L}(\mathbf{B})$. From $\mathbf{c}_i$, we obtain a the corresponding $\mathbf{a}_i \in \mathbb{Z}_q^n$, and since $\mathbf{c}_i$ is uniformly random, $\mathbf{a}_i$ is uniformly random in $\mathbb{Z}_q^n$. These vectors $\mathbf{a}_i$ are the input to SIS problem.

4. Applications

4.1. Public Key Encryption

With the spread of more insecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems. The process of encryption and decryption is depicted in the following illustration :

The most important properties of public key encryption scheme are:

- Different keys are used for encryption and decryption.This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plain text from the cipher text and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys

Some applications of SIS :
The SIS has the following properties and applications:
-Properties:

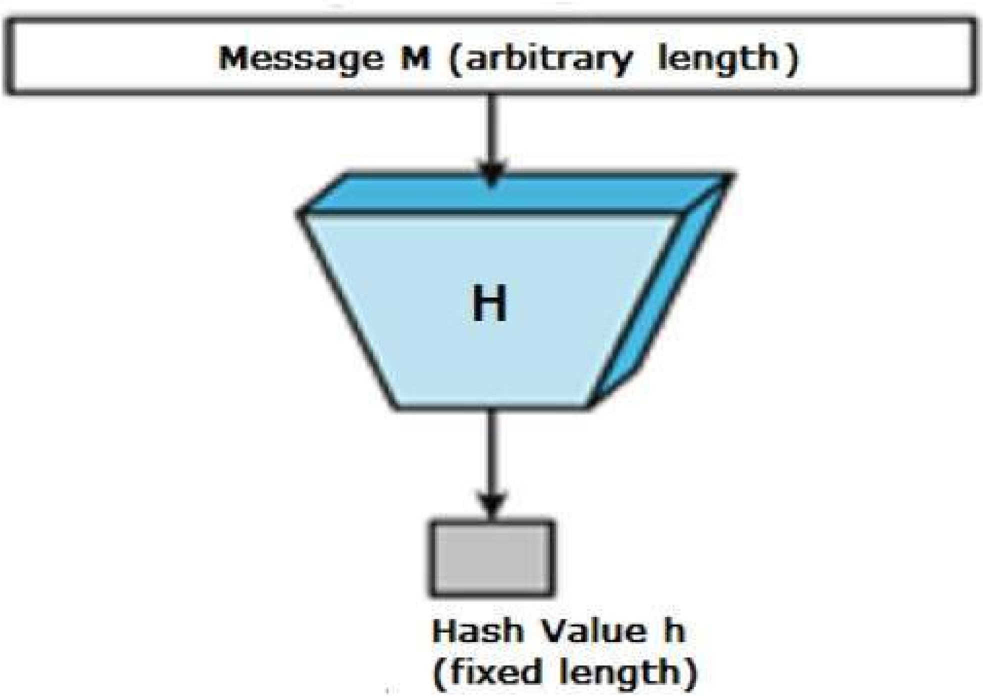
- Compression
- Regularity and homomorphism

-Applications:

- Collision Resistant Hashing
- Commitment Schemes
- Digital Signatures

4.2. Hash function

A hash function usually means a function that compresses, meaning the output is shorter than the input. Often, such a function takes an input of arbitrary or almost arbitrary length to one whose length is a fixed number, like 160 bits. Hash functions are used in many parts of cryptography, and there are many different types of hash functions, with differing security properties.



Properties:

In order to be an effective cryptographic tool, the hash function must possess the following properties :

- *Pre-Image Resistance*: This property means that it should be computationally hard to reverse a hash function. In other words, if a hash function h produced a hash value z, then it should be a difficult process to find any input value x that hashes to z. This property protects against an attacker who only has a hash value and is trying to find the input.

- *Second Pre-Image Resistance* This property means given an input and its hash, it should be hard to find a different input with the same hash. In other words, if a hash function h for an input x produces hash value $h(x)$, then it should be difficult to find any other input value y such that $h(y) = h(x)$. This property of hash function protects against an attacker who has an input value and its hash, and wants to substitute different value as legitimate value in place of original input value.

- *Collision Resistance*: This property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function. In other words, for a hash function

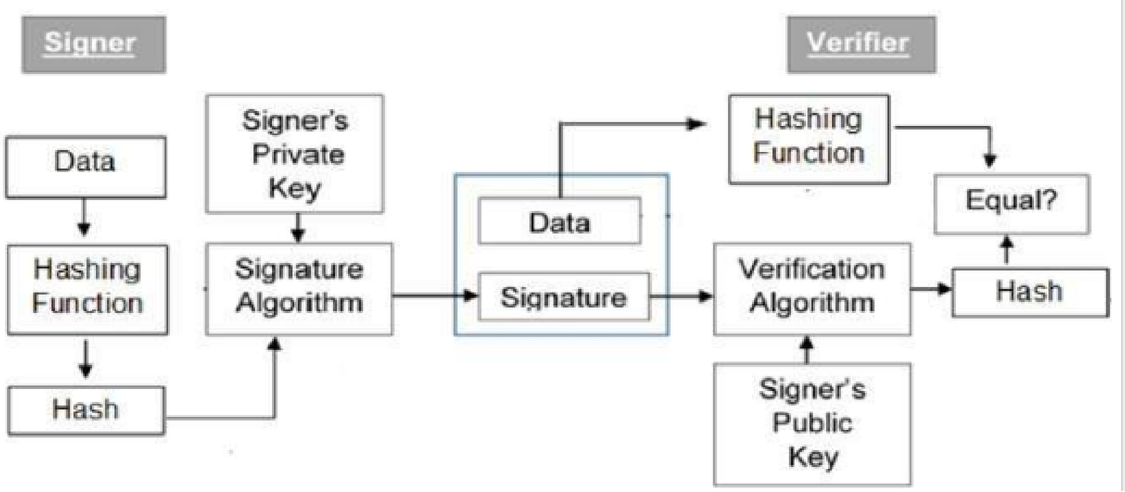
h, it is hard to find any two different inputs x and y such that $h(x) = h(y)$. Since, hash function is compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find.This property makes it very difficult for an attacker to find two input values with the same hash. Also, if a hash function is collision-resistant then it is second pre-image resistant.

4.3. Digital Signature

Digital signatures are the public-key primitives of message authentication. Similarly, a digital signature is a technique that binds a person/entity to the digital data.This binding can be independently verified by receiver as well as any third party. Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

In real world, the receiver of message needs assurance that the message belongs to the sender and he should not be able to repudiate the origination of that message. This requirement is very crucial in business applications, since likelihood of a dispute over exchanged data is very high.

Model of Digital Signature :
The digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration:



Explanation:

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.

- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.

- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.

- Verifier also runs same hash function on received data to generate hash value.

- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.

- Since digital signature is created by 'private' key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

4.3.1. Importance of Digital Signature

Out of all cryptographic primitives, the digital signature using public key cryptography is considered as very important and useful tool to achieve information security.

Apart from ability to provide non-repudiation of message, the digital signature also provides message authentication and data integrity. Let us briefly see how this is achieved by the digital signature.

- Message authentication - When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.

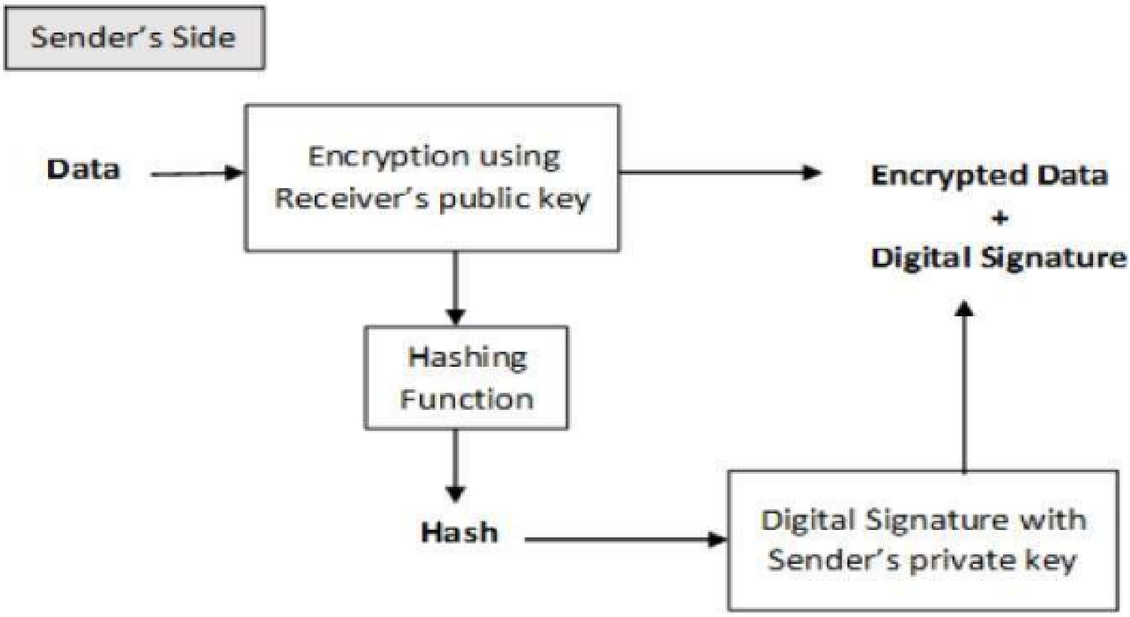
- Data Integrity - In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.

- Non-repudiation - Since it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future. By adding public-key encryption to digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security namely - Privacy, Authentication, Integrity, and Non-repudiation.

4.3.2. Encryption with Digital Signature:

In many digital communications, it is desirable to exchange an encrypted messages than plaintext to achieve confidentiality. In public key encryption scheme, a public (encryption) key of sender is available in open domain, and hence anyone can spoof his identity and send any encrypted message to the receiver.

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation.
This can archived by combining digital signatures with encryption scheme. Let us briefly discuss how to achieve this requirement. There are two possibilities, sign-then-encrypt and encrypt-then-sign. However, the cryptosystem based on sign-then-encrypt can be exploited by receiver to spoof identity of sender and sent that data to third party. Hence, this method is not preferred. The process of encrypt-then-sign is more reliable and widely adopted. This is depicted in the following illustration :


Sender's Side
Data
Encryption using Receiver's public key
Encrypted Data + Digital Signature
Hashing Function
Hash
Digital Signature with Sender's private key

The receiver after receiving the encrypted data and signature on it, first verifies the signature using sender's public key. After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.The receiver after receiving the encrypted data and signature on it, first verifies the signature using sender's public key. After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.

Acknowledgement. We would like to thank our supervisor, Dr.Hieu PHAN, for the patient guidance, encouragement and advice he has provided throughout our time. The meetings we had together helped us very much to understand well many points. We hope that our modest work will please you. Your time and consideration are greatly appreciated.

References

[1] A. Ajtai, *Generating hard instances of lattice problems*, Quaderni di Matematica, 13:1-32,2004. Preliminary version in STOC 1996.

[2] C. Peikert, *A Decade of Lattice Cryptography*

[3] D. Micciancio and O. Regev, *Worst-case to Average-case Reductions based on Gaussian Measures,* SIAM J.Comput., 37(1): 267-302,2007. Preliminary version in FOCS 2004.

[4] W. Banaszczyk, *New bounds in some transference theorems in geometry of numbers*, Mathematische Annalen, 296(4):625-635, 1993.

[5] https://www.cs.cornell.edu/courses/cs6830/2009fa/scribes/lecture21.pdf

[6] https://fr.wikipedia.org/wiki/Cryptographie_asym%C3%A9trique