Lattice Based Signatures with Additional Functionalities

Swati Rawal[1], Sahadeo Padhye[2] & Debiao He[3]

[1]*Department of Mathematics,*
Motilal Nehru National Institute of Technology,
Allahabad-211004, India.
[2] *EY Global, 6 More Londan Place, London-SE12AF, United Kingdom.*
[3] *School of Cyber Science and Engineering,*
Wuhan University, Wuhan, China
E-mail:swati.rawal25@gmail.com[1]
sahadeo@mnnit.ac.in[2]
hedebiao@whu.edu.cn[3]

Abstract

Digital signatures is a cryptographic protocol that can provide the added assurances of identity, status, proof of origin of an electronic document, and can acknowledge informed consent by the signer. Lattice based assumptions have seen a certain rush in recent years to fulfil the desire to expand the hardness assumption beyond factoring or discrete logarithm problem on which digital signatures can rely. In this article, we cover the recent progress made in digital signatures based on lattice assumptions. The article briefly discusses the working of each signature scheme, then investigates the progress made in recent years and compare them with different aspects of security and efficiency. Besides, it provides some future direction which can be helpful in future work in this area.

Keywords: Lattice based Cryptography, Group Signature, Proxy Signature, Ring Signature, Blind Signature, Attribute based Signature.

1. Introduction

A Digital Signature is an electronic data that confirms the identity of the sender and provides a layer of security to the messages sent through an insecure channel. It also includes non-repudiation, i.e., we have the assurance that the signer can't deny later that he is the signer of the message. Due to these features, digital signatures became the crux for software security, e-business, e-banking, online auctions, e-voting, e-cash, identity management, and many more applications.

As of yet, the security of many digital signatures is widely based on well-known number theoretical problems such as factoring and discrete-log problem. For decades, many algorithms were developed to solve these problems, but the fastest algorithms are sub-exponential on classical systems. However, Peter Shor [78, 79] gave an algorithm that can solve these problems in polynomial runtime on quantum computers. This development threatened the security of digital signature based on them; hard problems based lattices provide a concrete alternative that is secure against quantum attacks. Moreover, these schemes have an upper-hand over other areas as the security of these systems can be reduced to worst-case problems, and most of the schemes require simple computations to compute signatures instead of standard modular exponentiation.

Ajtai [5] connected worst-case problems and average-case problems for lattices, thus attracting many cryptographers to work on lattice-based schemes. Ajtai proved that certain problems are hard on average, provided if an underlying related problem on lattices is hard in worst-case. Such results become the foundations for the construction of many lattice based digital signature schemes. In this article, we try to survey lattice based digital signature schemes with a detailed signing protocol and security analysis. We proceed by defining the signature schemes with additional functionality in this section.

Blind Signature introduced in 1982 by Chaum [27] as the name suggested here the message is blind to the signer, i.e., the signer does not know the content of the message. It has applications where signer's privacy is essential such as e-voting, e-banking, etc.

Proxy Signatures was developed by Mambao et al. [61] in 1996 when the original signer is not available.

One or more (group) original signers delegates his/their signing rights to one or more proxy signers so that he/they can sign on his/their behalf.

Group Signature concept was given in 1991 by Chaum and Heyst [28], which allows any participant of the group of signers to sign anonymously on behalf of the group. It maintains anonymity as well as traceability, as only the group leader can trace the identity of the signer.

Ring Signature was introduced in 2001 by Rivest, Shamir, and Tauman [77]. Like group signature, here is also a single signer sign on behalf of the group, but here traceability feature is excluded; no one can trace the actual signer. Thus, this signature provides anonymity to the signer; it was introduced as a way to leak secrets so that the source can be authentic as well as anonymous.

Threshold Signature introduced by Y.Desmedt [31] in 1987. It facilitates the group to sign even if some members are not available. In a (t, n) threshold signature, at least t members of the group are required to generate a signature out of n members.

Attribute-based Signature introduced in 2011 by Maji et al. [60], where the signer satisfying a set of attributes can only obtain the signing rights and can sign the message with any predicates fulfilled by his attribute set. For example, suppose in a college we need the document to be signed by a professor in the maths department who is also a member of the hostel committee; we can use attribute signature to verify the attributes of the person who can sign the document.

The remaining segments in this paper are sorted out as follows. The second section begins with the basic preliminaries, and the preceding sections discuss the various basic signatures developed over lattices. Section four consists of signature schemes with additional functionalities, highlighting the recent development in the area along with the future direction. The last section concludes the paper.

2. Preliminaries

2.1. Notations

$\mathbb{R}$ and $\mathbb{Z}$ represents the set of reals and integers respectively. For any positive integer m, $\{1, 2, ...m\}$ is denoted by $[m]$. Extension of any real function $f()$ over a countable set A is $f(A) = \Sigma_{x\in A} f(x)$. Conventionally, vectors are written using bold lowercase letters $\mathbf{x}$ and are claimed to be in column form. Matrices are denoted by uppercase bold letters e.g. $\mathbf{X}$.

$\|\mathbf{x}\|$ is euclidean norm of vector $\mathbf{x}$. For any $n \times m$ matrix $\mathbf{A}$, we define $\|\mathbf{A}\| = max_{i\in[m]}\|\mathbf{x}_i\|$. If χ is any distribution then $s \leftarrow \chi$ means that s was sampled from a distribution χ. The statistical distance between any two distribution A and B is $\triangle\ (A, B) = \frac{1}{2}\Sigma_{x\in D}|A(x) - B(x)|$ over a countable domain D.

All through the paper, n is used as the natural security parameter, and standard big-O notation is used with its usual meaning. We say that $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n).log^c n)$, for some fixed integer c. A polynomial function in n, $poly(n)$ denotes an unspecified function $f(n) = O(n^c)$. A function $\epsilon(n)$ said to negligible if $\epsilon(n) = O(n^{-c})$ and we say an event occurs with overwhelming probability, if its probability is $1 - \epsilon(n)$, where $\epsilon(n)$ is a negligible function.

2.2. Lattices: Background and Definition

Definition 1. ***Lattice:*** *Consider $B = \{\mathbf{b_1}, \mathbf{b_2} \dots \mathbf{b_n}\}$ be the set of n linearly independent vectors in $\mathbb{R}^m$. Then, a lattice is defined by the set $\mathcal{L}(B) = \{\Sigma_{i=1}^n x_i\mathbf{b_i} | x_i \in \mathbb{Z}\}$. More conveniently, a lattice is the set of integer linear combinations of vectors of B.*

This set B can be represented as $n \times m$ matrix, and its columns form a basis for the lattice. If $n = m$, then the lattice is called a full rank lattice.

A lattice is also a discrete additive subgroup of $\mathbb{R}^n$. $\mathbb{Z}^n$ is the simplest example of a lattice. The shortest length of non-zero vectors in any lattice is defined to be the *minimum distance* of the lattice i.e. $\lambda_1(\mathcal{L}) := min_{v\in\mathcal{L}/\{0\}}\|v\|$. In general, ith successive minimum $\lambda_i(\mathcal{L})$ is defined to be the smallest radius r such that $\mathcal{L}$ has i linearly independent vectors of length atmost r.

Definition 2. ***Fundamental Domain:*** *Given a basis $\mathbf{B} = \{\mathbf{b_1}, \mathbf{b_2} \dots \mathbf{b_n}\} \in \mathbb{R}^{n\times m}$, the fundamental domain or parallelepiped is defined by $\mathcal{P}(B) = \{\Sigma_{i=1}^n x_i\mathbf{b_i} | x_i \in [0, 1)\}$.*

Note 1. *For every $x \in \mathbb{R}^m$, $\exists v \in \mathcal{L}$ such that $x = v + \mathcal{P}(\mathbf{B})$.*

Definition 3. ***Dual of Lattice:*** *The dual of any lattice $\mathcal{L}(B) \subset \mathbb{R}^n$ is $\mathcal{L}^\star = \{x | < x, \mathcal{L} > \in \mathbb{Z}\}$ (collection of all the points whose inner product with the vectors of the lattice is an integer).*

2.2.1. Hard computational problems on Lattices

We recall the computational problems over lattices that are of great importance in cryptography.

Definition 4. ***Shortest Vector Problem (SVP):*** *For the given lattice basis $\mathbf{B}$, find the shortest non-zero vector in $\mathcal{L}(\mathbf{B})$ i.e., find $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x}\| \leq \|\mathbf{y}\|$ for any other $\mathbf{y} \in \mathcal{L}(B)$*

Note 2. *A solution to SVP depends on the norm we are considering. In the same lattice if we consider two different norms, we may obtain two different shortest vectors.*

Definition 5. ***Closest Vector Problem*** *(CVP): Given a lattice $\mathcal{L}(\mathbf{B})$ with a basis $\mathbf{B}$ and a target vector $\mathbf{x}$, find the lattice vector $\mathbf{v} \in \mathcal{L}(B)$ closest to $\mathbf{x}$.*

Definition 6. ***Shortest Independent Vector Problem*** *(SIVP): For an $n-$dimensional lattice $\mathcal{L}(\mathbf{B})$, with a basis $\mathbf{B}$, find a set $S = \{\mathbf{s_i}\} \subset \mathcal{L}(\mathbf{B})$ of n linearly independent vectors such that $\|\mathbf{s_i}\| \leq \lambda_n(\mathcal{L})$.*

An important notion of lattice cryptography is the *approximation* problems parametrized by $\gamma(n) \geq 1$ (approximation parameter) which is the function of lattice dimension n. For the above problems approximate and decisional version is stated below.

Table 1: Approximation and Decisional Version of Lattice Problems

Problem	Approximation	Decisional
SVP	Given a basis $\mathbf{B}$, find a non-zero vector $x \in \lambda(\mathcal{L}(\mathbf{B}))$ such that $\|\mathbf{x}\| \leq \gamma(n)\lambda_1$.	$GapSVP_\gamma$: Given basis $\mathbf{B}$ and a positive integer d, then determine which is true : $\lambda(\mathcal{L}(\mathbf{B})) \leq \mathbf{d}$ or $\lambda(\mathcal{L}(\mathbf{B})) \geq \mathbf{d}\gamma(\mathbf{n})$.
CVP	Given a basis $\mathbf{B}$, and a target vector $\mathbf{x}$, find the lattice vector $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{x}\| \leq \gamma(n) dist(\mathcal{L}(\mathbf{B}), \mathbf{x})$.	$(GapCVP_\gamma)$: Given basis $\mathbf{B}$ and a positive integer d, then determine which is true : $dist(\mathcal{L}(\mathbf{B}), \mathbf{x}) \leq d$ or $dist(\mathcal{L}(\mathbf{B}), \mathbf{x}) \geq d\gamma(n)$.
SIVP	$SIVP_\gamma$: for an $n-$dimensional lattice $\mathcal{L}(\mathbf{B})$, with a basis $\mathbf{B}$, find a set $S = \{\mathbf{s_i}\} \subset \mathcal{L}(\mathbf{B})$ of n linearly independent vectors such that $\|\mathbf{s_i}\| \leq \gamma\lambda_n(\mathcal{L})$.	N.A.

There are other well studied problems also which have a significant influence over lattice cryptography, e.g., the *Hard-on average* problem introduced by Ajtai [5] known as *Smallest Integer Solution Problem*. Note that here we consider l_2- norm.

Definition 7. ***Smallest Integer Solution Problem*** *(SIS): For an integer q, a real β, and a matrix $\mathbf{A} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}\times\mathbf{m}}$, work out a integer vector (non-zero) $\mathbf{s} \in \mathbb{Z}^m$ ensuring that $\mathbf{As} = 0 \mod q$ and $\|\mathbf{s}\| \leq \beta$.*

There is a variant of this problem, where we find a solution to inhomogeneous system, as defined below.

Definition 8. ***Inhomogeneous Smallest Integer Solution Problem*** *(ISIS): For an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a syndrome $u \in \mathbb{Z}_q^n$, and a real β, workout the integer vector(non-zero) $\mathbf{s} \in \mathbb{Z}^m$ ensuring that $\mathbf{As} = u \mod q$ and $\|\mathbf{s}\| \leq \beta$.*

2.3. Discrete Gaussian

Gaussian-like probability distributions assume a significant job in lattice cryptography and are known as discrete gaussian. In this section we briefly talk about them.

Definition 9. ***Gaussian Function:*** *For $\alpha > 0$ a gaussian function $\rho : \mathbb{R}^n \longrightarrow \mathbb{R}^+$ centred at r is defined as*

$$\rho_{\alpha,r}(x) = exp(-\pi\|x - r\|^2/\alpha^2) \ , \ \forall x \in \mathbb{R}^n$$

Definition 10. ***Discrete Gaussian Distribution:*** *For some $r \in \mathbb{R}^n$, $\alpha > 0$, and an n-dimensional lattice $\mathcal{L}$, discrete gaussian distribution can be defined as*

$$D_{\mathcal{L},\alpha,r}(\mathbf{x}) = \frac{\rho_{\alpha,r}(\mathbf{x})}{\rho_{\alpha,r}(\mathcal{L})} \ , \ \forall \mathbf{x} \in \mathcal{L}$$

where, $\rho_{\alpha,r}(\mathcal{L}) = \Sigma_{\mathbf{y}\in\mathcal{L}}\rho_{\alpha,r}(\mathbf{y})$.

Micciancio and Regev [64] gave a fascinating lattice quantity called *smoothing parameter*.

Definition 11. ***Smoothing Parameter:*** *For a positive real $\epsilon > 0$, n-dimensional lattice $\mathcal{L}$, the smoothing parameter $\eta_\epsilon(\mathcal{L})$ of $\mathcal{L}$ is the smallest real $\sigma > 0$ such that $\rho_{1/\sigma}(\mathcal{L}^*/0) \leq \epsilon$.*

In an informal manner, the smoothing parameter is defined as the amount of *blur* required to smooth out the discrete structure of the lattice.

Note 3. *Many lattice quantities are closely related to the smoothing parameter [9, 38, 64].*

1. $\eta_{2^{-n}}(\mathcal{L}) \leq \sqrt{n}\ \lambda_1(\mathcal{L}^\star)$ *, for any n-dimensional lattice $\mathcal{L}$.*
2. *For $\epsilon \in (0, 1/2)$, and any n-dimensional lattice $\mathcal{L}$,*

$$\eta_\epsilon(\mathcal{L}) \leq \min_{(basis\ \mathbf{B}\ of \mathcal{L})} \|\tilde{\mathbf{B}}\| \sqrt{\log O(n/\epsilon)} \leq \lambda_n(\mathcal{L}) \sqrt{\log O(n/\epsilon)}.$$

2.4. Learning with Error (LWE)

Regev [74] in 2005 introduced a new *average-case problem* which is coined as an "encryption-enabled" analogue of SIS problem, known as *learning with error*. Positive integers n, q, and an error distribution χ over $\mathbb{Z}$ are main parameters of LWE. n, q are taken roughly the same as in SIS, and the error distribution χ is the discrete Gaussian with the width αq for some $\alpha < 1$, which is known as relative "error rate."

Definition 12. ***LWE Distribution:*** *For a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathcal{A}_{s,\chi} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, the LWE distribution is sampled by choosing uniformly at random $\mathbf{a} \in \mathbb{Z}_q^n$, $e \leftarrow \chi$, and returning $(\mathbf{a}, \mathbf{b} =< \mathbf{s}, \mathbf{a} > + e) \mod q$.*

There are other versions of the LWE problem also, search and decisional LWE. In *search* LWE, the main aim is to output the secret vector from the given LWE samples. And in the *decisional* LWE, we need to differentiate LWE samples from uniformly random samples. These problems have an additional parameter, m the number of available samples, which is usually large enough so that with high probability, the secret is uniquely defined.

Definition 13. ***Search LWE:*** *For m independent LWE samples $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ sampled from LWE distribution $\mathcal{A}_{s,\chi}$, and a fixed uniformly random secret $\mathbf{s} \in \mathbb{Z}_q^n$, output $\mathbf{s}$.*

Definition 14. ***Decisional LWE:*** *For m independent samples $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, distributed according to either $\mathcal{A}_{s,\chi}$ for a uniformly random secret $\mathbf{s} \in \mathbb{Z}_q^n$ (fixed for all samples), or the uniform distribution, differentiate them and output the case with non-negligible advantage.*

Note 4.

1. *The above-defined problems are easy to solve in the absence of the error term $e \leftarrow \chi$, as we can efficiently solve for secret $\mathbf{s}$ from the given LWE samples by Gaussian elimination.*
2. *Similar to SIS problem, to our convenience we combine the given LWE samples into a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, whose columns are the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$, and a vector $\mathbf{b} \in \mathbb{Z}_q^m$ comprising of vectors $\mathbf{b}_i \in \mathbb{Z}_q$, so that we have*

$$\mathbf{b} = \mathbf{A}^T\mathbf{s} + e\ ,\ e \leftarrow \chi^m$$

In the case of decisional LWE, $\mathbf{b}$ is uniformly random and independent of $\mathbf{A}$.

For the hardness of LWE, Regev [74] stated and proved the following theorem to show that its hardness is equivalent to solving $GapSVP_\gamma$ or $SIVP_\gamma$.

Theorem 1. *[74]For a security parameter n, choose $m = poly(n)$ and modulus $q \leq 2^{poly(n)}$. With parameter $\alpha q \geq 2\sqrt{n}$, $0 < \alpha < 1$, a discrete Gaussian error distribution χ is selected, then solving the decisional problem is at least as hard as quantumly solving $GapSVP_\gamma$ and $SIVP_\gamma$ on arbitrary n-dimensional lattices, for some $\gamma = \tilde{O}(n/\alpha)$.*

3. Basics Signature Schemes over Lattices

In this section, we discuss the basic signature schemes over lattices. These schemes were used as primary building blocks for the signature schemes with additional functionalities. Before proceeding further, we define a formal structure of a digital signature scheme.

Definition 15. ***Digital Signature Scheme.*** *It consists of four algorithms discussed below.*

- $Setup(\lambda)$: *An algorithm which takes as input a security parameter λ, and outputs the public parameters params.*
- $KeyGen(params)$: *Generates the signing key or secret key sk, and the public verification key pk for given parameters.*
- $SignGen(m, sk, params)$: *An algorithm which takes as input the message m, a signing key sk, and the public parameters params, outputs the signature σ on m.*
- $Verification(params, vk, m, \sigma)$: *This is a deterministic algorithm that returns 1 if the signature is valid else 0 if invalid.*

We classify a digital signature scheme to be secure if it is existentially unforgeable under the chosen message attack. Now, moving towards the signature schemes constructed over lattices, the very first discussion is on the signature scheme using trapdoor.

3.1. Signature Scheme Using Trapdoor

This scheme, at its core, depends on the trapdoor function used. It utilizes a family of special one-way and collision resistant trapdoor functions, known as *Preimage sampleable function*. Along with these functions and hash-sign paradigm [32], Gentry, Peikert and Vaikuntanathan [38] gave the signature scheme. Now, we define these functions, followed by their construction.

Definition 16. ***Preimage Sampleable Functions:*** *These functions are defined by a quadruple of probabilistic polynomial-time algorithms* $(TrapGen, SampD, SampDom, SampPre)$

- $TrapGen(1^n)$: *This algorithm generates a function with a trapdoor. With security parameter as an input, it outputs* (a, t)*, where a defines a function(efficiently computable)* $f_a : D^n \longrightarrow R^n$ *with domain* D^n *and range* R^n*, and a trapdoor t.*
- $SampD(1^n)$: *An algorithm that outputs samples from a discrete Gaussian defined over the lattice.*
- $SampDom(1^n)$: *This algorithm outputs a sample* $\mathbf{x}$ *from some distribution defined over* D^n*, such that the distribution of* $f_a(\mathbf{x})$ *over* R^n *is uniform.*
- $SampPre(t, \mathbf{y})$: *For every* $\mathbf{y}$*,* $SampPre(t, \mathbf{y})$ *samples from the conditional distribution of* $\mathbf{x} \longrightarrow SampDom(1^n)$*, given* $f_a(\mathbf{x}) = \mathbf{y}$.

Note 5. *The probability that* $\mathcal{A}(1^n; a; \mathbf{y}) \in {f_a}^{-1}(\mathbf{y}) \subseteq D^n$ *, for any probabilistic polynomial-time algorithm* $\mathcal{A}$*, is negligible.*

3.1.1. Construction of Preimage sampleable function

First, recalling the result in [6], that states how we can sample an essentially uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ along with a relatively short full-rank trapdoor set of lattice vectors $\mathbf{S} \subset \mathcal{L}^{\perp}(\mathbf{A})$.

Lemma 1. *[6] For a given security parameter n, a prime* $q = poly(n)$ *and any* $m \geq 5n \log q$*, there exists a probabilistic polynomial-time algorithm that, outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *such that its distribution is statistically close to uniform over* $\mathbb{Z}_q^{n\times m}$ *and a full-rank set of lattice vectors* $\mathbf{S} \subset \mathcal{L}^{\perp}(\mathbf{A})$*, where* $\|\mathbf{S}\| \leq L = m^{2.5}$

There is another interesting result [63] stating that this set $\mathbf{S}$ can be converted to a "good" basis $\mathbf{T}$ such that $\|\tilde{\mathbf{T}}\| \leq \|\tilde{\mathbf{S}}\| \leq L$.

Lemma 2. *[63] For an arbitrary basis* $\mathbf{B}$ *of an n-dimensional lattice* $\mathcal{L}$ *and a full-rank set* $S \in \mathcal{L}$ *of lattice vectors, there is a deterministic polynomial-time algorithm that outputs a basis* $\mathbf{T}$ *such that* $\|\tilde{\mathbf{t}}_i\| \leq \|\tilde{\mathbf{s}}_i\|$ *,* $\forall i \in [n]$.

With the help of the above results, we present the construction of PSF's based on the average-case hardness of SIS and/or ISIS. Here the system parameters are taken according to the above proposition along with the gaussian parameter $s \geq L\omega(\sqrt{\log m})$.

- $TrapGen$: Use Lemma 2 to generate $(\mathbf{A}, \mathbf{T})$, where $\mathbf{A}$ defines a function $f_{\mathbf{A}}$, and the good basis $\mathbf{T}$ forms its trapdoor.
- The function $f_{\mathbf{A}}$ is defined as $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{Ae} \mod q$, with a domain $D^m = \{\mathbf{e} \in \mathbb{Z}^m : \|\mathbf{e}\| \leq s\sqrt{m}\}$, and range $R^n = \mathbb{Z}_q^n$. The distribution of the domain is $D_{\mathbb{Z}^m,s}$, sampled using $SampD$ utilizing standard basis for $\mathbb{Z}^m$.
- $SampleISIS(\mathbf{A}, \mathbf{T}, s, \mathbf{u})$ (trapdoor-inversion algorithm) samples from ${f_{\mathbf{A}}}^{-1}(\mathbf{u})$, it first chooses an arbitrary $\mathbf{t} \in \mathbb{Z}^m$ satisfying $\mathbf{At} = \mathbf{u} \mod q$. Then, samples $\mathbf{v} \sim D_{\mathcal{L}^{\perp},s,-t}$ using $SampD(\mathbf{T}, s, -\mathbf{t})$, and output $\mathbf{e} = \mathbf{t} + \mathbf{v}$.

Note 6. *In the above construction, the existence of* $\mathbf{t}$ *is guaranteed from the following lemma.*

Lemma 3. *[5] For* $m \geq 2n \log q$*. Then for all but for at most* q^{-n} *fraction of* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$*, for every syndrome* $\mathbf{u} \in \mathbb{Z}_q^n$*,* $\exists\, \mathbf{e} \in \{0, 1\}^n$ *such that* $\mathbf{Ae} = \mathbf{u} \mod q$.

3.1.2. Signature Scheme

The Signature scheme is constructed using the above a collection of collision-resistant PSFs and a hash function $H : \{0, 1\}^{\star} \longrightarrow R^n$ (used as a random oracle).

- $KeyGen(1^n)$: Generate $(a, t) \leftarrow TrapGen(1^n)$, where a (verification key) defines f_a, and trapdoor of f_a is t which is the signing key.
- $SignGen(t, m)$: Generate $\sigma_m \leftarrow SampPre(t, H(m))$, and output the signature σ_m.
- $Verify(a; m; \sigma_m)$: Signature is valid if $\sigma_m \in D^n$, and $f_a(\sigma_m) = H(m)$ else it is invalid.

3.1.3. Security Analysis

The above scheme is proven strongly existentially unforgeable against chosen-message attack [38] under the hardness of SIS problem. It is proved (refer proposition 6.1 of [38]) that if there exists any successful forger, then using a valid forgery we can construct a successful attacker against SIS problem. Thus, the scheme is secure under the hardness of SIS problem.

Remark 1. *In the recent development, a new method of trapdoor generation was developed by Micciancio et al. .[65] in 2011, which is simple, efficient, facile to implement, and is optimized with a small constant in comparison to previous algorithms [6, 7]. Along with trapdoor generation, they also introduced more efficient trapdoor inversion and pre-image sampling, which provides an efficient and more practical setting for schemes such as [8, 38, 43, 70]. They also gave a detailed comparison of their construction with the prior ones.*

In addition to the signature scheme with trapdoor, as discussed above, the signature schemes without trapdoor exist in the literature, which turns out to be more efficient and providing shorter keys & signatures.

3.2. Signature scheme without trapdoor

The signature scheme without using any trapdoor function, based on lattices was introduced in [58]. The scheme employed *rejection sampling* during signature generation to get the desired results. The method of rejection sampling is discussed in the following subsection.

3.2.1. Rejection Sampling

Von Neumann [67] introduced the method of *rejection sampling* in 1951. The key idea of the method is to output samples from a probability distribution f using the samples from the probability distribution g. The samples from g are accepted with probability $f(x)/Mg(x)$, and the process is repeated until the sample is accepted.

Here, we have to keep M as small as possible as it represents the expected number of times repetition occurs. See Figure 1, the grey shaded region defines the rejection area and if (x_i, y_i) is the sample then it is accepted if $y_i \leq f(x_i)$.

Note 7. *Taking g as close as possible to f, reduced the factor of repetition(M).*

This method of rejection sampling is stated as the following lemma, which forms the key ingredient of the signature scheme proposed in [58].

Lemma 4. *[58] For an arbitrary set V, define probability distributions $h : V \longrightarrow \mathbb{R}^n$ and $f : \mathbb{Z}^m \longrightarrow \mathbb{R}$. Now, we define a family of probability distributions indexed by all $v \in V$ satisfying a property that $\exists M \in \mathbb{R}$, such that $\forall v$, $Pr[Mg_v(z) \geq f(z) : z \leftarrow f] \geq 1 - \epsilon$ then consider the following algorithm $\mathcal{A}$:*

1. $v \leftarrow h$
2. $z \leftarrow g_v$
3. *output* (z, v) *with probability* $min(\frac{f(z)}{Mg_v(z)}, 1)$.

Its output is within statistical distance $\frac{\epsilon}{M}$ *from the output of following algorithm* $\mathcal{F}$

1. $v \leftarrow h$
2. $z \leftarrow f$
3. *output* (z, v) *with probability* $\frac{1}{M}$.

Moreover, the output of algorithm $\mathcal{A}$ *occurs with the probability atleast* $\frac{1-\epsilon}{M}$.

The following signature scheme was constructed under the hardness of SIS problem using rejection sampling. After presenting the construction, we describe how this rejection works during signature generation.

3.2.2. Signature Scheme

The signature scheme employs a random oracle $H : \{0,1\}^\star \longrightarrow \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$. Its security based on the average-case hardness of $SIS_{q,n,m,\beta}$, where $\beta = \tilde{O}(n)$. It consists of three-tuple $(KeyGen, SignGen, Verify)$.

- $KeyGen(1^n)$: With the input of security parameter, it outputs the keys, the signing key $\mathbf{S} \in \{-d, \ldots, 0, \ldots d\}^{m \times k}$ and the public (verification) keys consist of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} \in \mathbb{Z}_q^{m \times k} = \mathbf{AS}$.
- $SignGen(\mu, \mathbf{A}, \mathbf{S})$:
  - (i) To sign on any message μ, signer first chooses a vector $\mathbf{y}$ from D_σ^m.
  - (ii) Now it computes $\mathbf{c} = H(\mathbf{Ay}, \mu)$ and then set $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$.
  - (iii) It outputs the signature $(\mathbf{z}, \mathbf{c})$ with probability min $(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1)$
- $Verify((\mathbf{z}, \mathbf{c}), \mu, \mathbf{A}, \mathbf{T})$: This algorithm returns 1 if $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ and $\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc}, \mu)$, else it return 0.

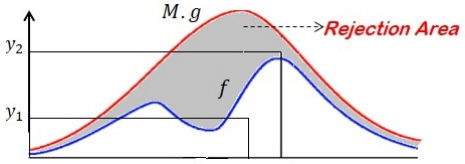


Figure 1: Rejection sampling : Sampling from g to get sample from f

How the rejection sampling came into the picture and what is its role in the above algorithm? Notice that the signature $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$ has its distribution dependent on the distribution of $\mathbf{Sc}$ or in fact on the distribution of $\mathbf{S}$. Moreover, it is nothing but the distribution D^m_σ shifted by $\mathbf{Sc}$ vector. Now, here comes the role of rejection sampling, it removes this dependency on the distribution of S. As we want our signature to be from the distribution D^m_σ , but we are getting it from the distribution $D^m_{\sigma,\mathbf{Sc}}$ and this is the purpose of the rejection sampling. The following lemma talks about the success probability of the rejection sampling.

Lemma 5. *[58]* $Pr[D^m_\sigma(\mathbf{z})/D^m_{\mathbf{x},\sigma}(\mathbf{z}) = O(1); \mathbf{z} \leftarrow D^m_\sigma] = 1 - 2^{-\omega(log(m))}$*, if* $\sigma = \omega(\|\mathbf{x}\|\sqrt{\log m})$ *and* $\mathbf{x} \in \mathbb{Z}_m$*. In fact,* $Pr[D^m_\sigma(\mathbf{z})/D^m_{\mathbf{x},\sigma}(\mathbf{z}) < e^{12/\alpha+1/(2\alpha^2)}; \mathbf{z} \leftarrow D^m_\sigma] > 1 - 2^{-100}$ *, for* $\sigma = \alpha\mathbf{Sc}$ *and* $\mathbf{x} \in \mathbb{Z}_m$.

3.2.3. Secuirty Ananlysis

The security of the above signature scheme is based on the hardness of the SIS problem under the random oracle. If any successful forger exists, then using valid forgery, we can construct a polynomial time algorithm that can find collisions for the hash function used or can solve the SIS problem. This result is stated as the theorem below, and the detailed proof can be referred from [58].

Theorem 2. *[58] If there is a polynomial-time forger who can break the above presented scheme with probability* ϵ *who can make atmost* s *and* h *sign queries and hash queries respectively then we can construct a polynomial-time algorithm that can solve the* $l_2 - SIS_{q,m,n,\beta}$ *for* $\beta = (2\eta\sigma + 2d\kappa)\sqrt{m}$ *with probability* $\approx \dfrac{\epsilon^2}{2(h+s)}$.

The next section discusses the signature scheme popularly known as BLISS, which was proposed as an improvement of the above scheme. The following scheme improved the signature size and the efficiency making it more convenient for practical implementation.

3.3. Lattice Signatures and Bimodal Gaussian

This signature scheme was introduced by Leo Ducas et al. [33] in 2013 as an improvement of the above scheme. Heart of the Lyubashevsky scheme [58] was its rejection sampling algorithm. The new scheme [33] improved this method itself, which not only resulted in computational efficiency but enhanced security and small key sizes. They also discussed its implementation and compared its efficiency for security level 128 bits, 160 bits, and 192 bits with the existing RSA and ECDSA.

Now to understand how this new scheme [58] emerged as an improvement of the above scheme, in the previous scheme we were sampling the signature $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$, but here we first uniformly choose a bit from $\{-1, 1\}$ and then generate the signature as $\mathbf{z} = b\mathbf{Sc} + \mathbf{y}$, thus the signature is sampled from $\frac{1}{2}D^m_{\mathbf{Sc},\sigma} + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}$. See the Figure 2 [33].

Figure-2(a) represents the sampling from the previous scheme, here D^m_σ (dashed red curve) is scaled by a large factor so that it can fit under it, but this increase the rejection area. Whereas in Figure-2(b), D^m_σ fits much better under the bimodal distribution and the rejection area is also reduced to a large extent. Due to the use of bimodal distribution in the scheme, this signature scheme is commonly referred to as BLISS (Bimodal Lattice Signature Scheme). In the next section, we discuss how verification is changed to adapt this bimodal distribution in the signature scheme.

3.3.1. Signature Scheme

The signature scheme is based on the hardness of $SIS_{q,n,m,\beta}$ and consists of a hash function H modelled as a random oracle with range $\mathbb{B}^n_\kappa$, set of binary vectors with length n and weight κ.

- *KeyGen* : Similar to the previous scheme generate a short matrix which serve as a secret key $\mathbf{S} \in \mathbb{Z}^{m\times n}_{2q}$ and the public key $\mathbf{A} \in \mathbb{Z}^{n\times m}_{2q}$ such that $\mathbf{AS} = \mathbf{A}(-\mathbf{S}) = q\mathbf{I}_n \mod 2q$.

- $SignGen(\mu, \mathbf{A}, \mathbf{S}, \sigma \in \mathbb{R})$: To get signature on μ

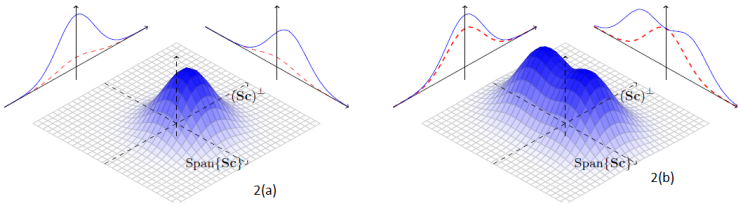


Figure 2: Rejection Sampling

repeat the compute $\mathbf{A}$ and $\mathbf{S}$ in same manner as in the previous scheme

(i) We have $\mathbf{c} = H(\mathbf{Ay}, \mu)$, now choose a random bit $b \in \{0, 1\}$.

(ii) Compute $\mathbf{z} = (-1)^b\mathbf{Sc} + \mathbf{y}$ and output the signature $(\mathbf{z}, \mathbf{c})$

The above algorithm generates the output with probability $\frac{1}{M}\exp(-\frac{\|b\mathbf{Sc}\|^2}{2\sigma^2})cosh(\frac{<\mathbf{z}, \mathbf{Sc}>}{2\sigma^2})$.

- $Verify((\mathbf{z}, \mathbf{c}), \mu, \mathbf{A})$: This algorithm either return accepted or rejected and works as follows :

(i) Check if $\|\mathbf{z}\| \leq B_2$ else return rejected.

(ii) Check if $\|\mathbf{z}\|_\infty < q/4$ else return rejected.

(iii) Verify whether $\mathbf{c} = H(\mathbf{Az} + q\mathbf{c}, \mu)$

Here, the bound B_2 is assumed higher than $m\sqrt{\sigma}$, as the signature we are expecting should be from D^m_σ. The second condition is imposed due to technical reasons in the security proof.

Now, concerning about the security, it was proved in [33] that any successful existential forgery results in the solution of the $SIS_{q,m,n,\beta}$ problem.

Theorem 3. *[33] For a polynomial-time forger $\mathcal{F}$ which can query signing oracle as well as random oracle H, s and h times respectively, if $\mathcal{F}$ can output a forgery with non-negligible probability ϵ, then there is a polynomial-time algorithm $\mathcal{A}$ that can solve $SIS_{q,m,n,\beta}$ where, $\beta = 2B_2$ with probability* $\approx \frac{\delta^2}{2(h+s)}$

3.3.2. Security Ananlysis

The hardness of the SIS problem and the random oracle guarantees the security of the scheme [33]. The security of the above-discussed signature schemes is based on the random oracle model. The security analysis states that if there is a successful forger, then using the valid forgery can solve the SIS problem, thus reducing the security to the SIS problem's hardness.

The next section discusses the signature scheme using trapdoor and security proved in the standard model. The following signature scheme is used to construct many signature schemes, such as proxy signature with additional functionalities.

3.4. Lattice signature using Bonsai Tree

David Cash et al. [24] introduced a new *hash-sign* scheme in the standard model. They proposed a new concept called *Bonsai trees*. Using this concept, they introduced a *Stateless* signature scheme whose security is based on the hardness of the SIS problem. It also uses a *chameleon* hash function. This scheme has a drawback over other schemes like GPV [38] that the public key size increases by a factor k of the output size of the chameleon hash function. Now, first introducing the chameleon hash function.

Definition 17. ***Chameleon Hash function :*** *It was introduced by Krawczyk and Rabin [44] in 2000. The family of such hash functions is a collection $\mathcal{H} = \{h_i : \mathcal{M}\times\mathcal{R} \longrightarrow \mathcal{Y}\}$ of efficiently computable h_i's which maps $m \in \mathcal{M}$(message), $r \in \mathcal{R}$(randomness) to a member of $\mathcal{Y}$ such that the tuple $(h_i, h_i(m, r))$ has uniform distribution upto a negligible statistical distance over $(\mathcal{H}, \mathcal{Y})$.*

The chameleon property of above collection is that a random h_i may be generated along with a trapdoor t such that it is possible to sample a randomness $r \in \mathcal{R}$ satisfying $h_i(m, r) = y$ for given output $y \in \mathcal{Y}$ and message $m \in \mathcal{M}$. The family also satisfy the collision-resistant property.

3.4.1. Bonsai trees and their growth principles

In a cryptographical sense, the tree is a hierarchy of the trapdoor function. In undirected growth, the cultivator doesn't have information about the associated function, whereas, in controlled growth, he knows the trapdoor. In the cryptographical bonsai tree, this controlled growth travel down to the children, i.e., if we know the

parent's trapdoor, then this implies that we know the trapdoor for children also.

Above was just an overview of the concept of a bonsai tree. Now we lay out the main techniques that we require. It has four basic principles, undirected growth, controlled growth, extending controlled growth over new growth and randomize growth.

Definition 18. ***Undirected Growth :*** *This growth helps us to infuse a challenging hard problem like SIS or LWE into the tree. We form a parity check matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *by grouping uniformly independent samples* $\mathbf{a_i} \in \mathbb{Z}_\mathbf{q}^\mathbf{n}$. *We set* $\mathbf{A}' = \mathbf{A}\|\overline{\mathbf{A}} \in \mathbb{Z}_q^{n\times m'}$, $m' > m$. *Then* $\mathcal{L}^\perp(\mathbf{A}') \subset \mathbb{Z}^{m'}$ *is a higher dimensional superlattice of* $\mathcal{L}^\perp(\mathbf{A}) \subset \mathbb{Z}^m$. *Then for any* $\mathbf{v} \in \mathcal{L}^\perp(\mathbf{A})$, *the vector* $\mathbf{v}' = \mathbf{v}\|0 \in \mathbb{Z}^{m'} \in \mathcal{L}^\perp(\mathbf{A}')$ *because* $\mathbf{A}'\mathbf{v}' = \mathbf{A}\mathbf{v} = 0 \in \mathbb{Z}_q^n$.

Definition 19. ***Controlled Growth :*** *In controlled growth we know the good or short (relatively) basis for a lattice.*

Following lemma help us to generate a family of lattices in a controlled manner.

Lemma 6. *[7] For a security parameter n, a fixed* $C > 0$, *and a given probabilistic polynomial-time algorithm* $GenBasis(1^n, 1^m, q)$, *where* $m > Cn\log q$, *returns* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *having a uniform distribution within a negligible statistical distance and a basis* $\mathbf{S} \in \mathbb{Z}_q^{m\times m}$ *of lattice* $\mathcal{L}^\perp(\mathbf{A})$ *such that* $\|\tilde{\mathbf{S}}\| \le \tilde{L} = O(\sqrt{n\log q})$.

Definition 20. ***Extending Controlled Growth :*** *There is a deterministic polynomial-time algorithm* $ExtBasis(\mathbf{S}, \mathbf{A}' = \mathbf{A}\|\overline{\mathbf{A}})$, *for a given* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a basis* $\mathbf{S} \in \mathbb{Z}^{m\times m}$, *returns the basis* $\mathbf{S}'$ *of* $\mathcal{L}^\perp(\mathbf{A}') \subset \mathbb{Z}^{m+m'\times m+m'}$ *such that* $\|\tilde{\mathbf{S}}\| = \|\tilde{\mathbf{S}}'\|$

Definition 21. ***Randomizing Control Growth :*** *Here, the cultivator randomizes the control, i.e., lattice basis with a little-bit loss in quality. We describe* $RandBasis(\mathbf{S}, s)$ *probabilistic polynomial-time algorithm which randomize the control.*

$RandBasis(\mathbf{S}, s)$ takes input as basis $\mathbf{S}$ of an m-dimensional lattice $\mathcal{L}$ and the parameter s such that $s \ge \|\tilde{\mathbf{S}}'\|\omega(\sqrt{\log n})$ and then returns basis $\mathbf{S}'$ of $\mathcal{L}$. The algorithm works as follows :

1. For $i = 0$, while $i < m$,
   choose $\mathbf{x} \leftarrow SampleD(\mathbf{S}, s)$, If $\{\mathbf{x}_1, \dots \mathbf{x}_i\}$, then let $i \leftarrow i + 1$ and $\mathbf{x}_i = \mathbf{x}$.

2. Return $\mathbf{S}' = ToBasis(\mathbf{V}, HNF(\mathbf{S}))$. (Here Hermite normal form of $\mathbf{S}$ is used so that no information is leaked about $\mathbf{S}$).

3.4.2. Signature Scheme

The signature scheme using this bonsai tree uses following parameters for security parameter n.

1. $m = O(n\log q)$ (dimension) and bound $\tilde{L} = O(n\sqrt{\log q})$.

2. Length of the hashed message is k, which changes the dimension m to $m.(k + 1)$

3. $s = \tilde{L} = O(n\omega(\sqrt{\log n}))$ be the Gaussian parameter.

- $KeyGen$: Using $GenBasis(1^n, 1^m, q)$, generate $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and basis $\mathbf{S}_0 \in \mathcal{L}^\perp(\mathbf{A}_0)$ such that $\|\tilde{\mathbf{S}}_0\| \le \|\tilde{L}\|$.
  Now, for each $(b, j) \in \{0, 1\} \times [k]$, choose ${\mathbf{A}_j}^b \in \mathbb{Z}_q^{n\times m}$ uniformly random and independent. Then the Public key is $vk = (\mathbf{A}_0, \{{\mathbf{A}_j}^b\})$, and the signing key is $(\mathbf{S}_0, vk)$.

- $SignGen(sk, M \in \{0, 1\}^k)$: Compute $\mathbf{A}_M = \mathbf{A}_0\|{\mathbf{A}_1}^{(M_1)}\| \dots \|{\mathbf{A}_k}^{(M_k)} \in \mathbb{Z}_q^{n\times m'}$. Now compute $\mathbf{v} \leftarrow SampD(ExtBasis(\mathbf{S}_0, \mathbf{A}_M), 0, s)$ distributed according to $D_{\mathcal{L}^\perp(\mathbf{A}_M),s}$.
  Return $\mathbf{v}$ as a signature on M.

- $Verify(\mathbf{v}, M, vk)$: Accept the signature iff $\mathbf{v} \ne 0$, $\|\mathbf{v}\| \le s\sqrt{m'}$, and $\mathbf{A}_M\mathbf{v} = 0$.

The above scheme gave a new method for extending the basis of one known lattice to its extended lattices. This new methodology can be efficiently used in signature schemes with additional functionalities. The following theorem proved existentially unforgeable of the above scheme under the chosen message attack.

Theorem 4. *[24] For any successful forger* $\mathcal{F}$, *mounting chosen-message attack on the above signature scheme, with atmost* q_s *sign queries, there exists an attacker against the* $SIS_{q,\beta}$ *problem.*

The above-discussed schemes form the main building blocks of the schemes discussed in the next section, signature schemes with additional functionalities.

4. Signature Schemes With Additional Functionalities

Till now, we have discussed the basis signature schemes that were developed over lattices. In this section, we proceed towards signature schemes that has some additional features. The first scheme discussed in this section is the *Group Signature* scheme developed by Gordon et al. [37]. Group Signature scheme combines two main properties *anonymity* and *traceability*. It

has many real-world applications. Suppose a large company wants every product to have a digital signature of the company to resist piracy. It is not always possible for the owner to sign every product; in this situation, the group signature can be life-saving. The owner can act as a group leader and assign signing rights to different heads in the company. Thus, every product is signed and verified as signed by the company; the owner can trace the signer in case of any dispute. It also has many other applications like privacy-protecting protocols, keycard access, auction protocol, etc.

4.1. Group Signature Scheme

Group Signature was first introduced in [28], here a member of a group signs on behalf of the group without revealing his identity to the verifier; only the group leader can trace him in case of any dispute. The Formal Structure of this scheme is adopted from the definition given by Bellare et al. [13] with the relaxations suggested by Boneh et al. [17]. This scheme is a quadruple of polynomial-time algorithms $(G\_KeyGen, G\_Sign, G\_Verify, G\_Open)$ defined as follows :

- $G\_KeyGen(1^n, 1^N)$: It takes as an input the security parameter n, the group size N and returns the N signing keys $sk[i]$ and verification key vk for the group.
- $G\_Sign(sk[i], M)$: It returns the signature σ on the input message M using the signing keys $sk[i]$.
- $G\_Verify(\sigma, vk)$: It outputs 1 or 0 that indicates accepted or rejected respectively.
- $G\_Open(tk)$: This is a tracing algorithm, it determines the signer who signed the message M.

The group signature scheme, as discussed before, has two essential features, anonymity and traceability. Anonymity property resists the tracing of the particular signer who signed the message given the public keys, and by traceability, the group leader can identify the signer.

This signature scheme, along with $TrapGen$ algorithm, uses another algorithm $orthoSample$, which takes an additional input, a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, and returns $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and its trapdoor $\mathbf{T} \in \mathbb{Z}^{m\times m}$ such that the rows of the output matrix $\mathbf{A}$ are orthogonal to the rows of the input matrix $\mathbf{B}$ ($\mathbf{AB}^T = \mathbf{0} \mod q$). The algorithm is stated as the following lemma.

Lemma 7. *[37] There is an algorithm* $orthosample(1^n, 1^m, q, \mathbf{B})$ *(* $q \geq 2, m \geq n + 8n\log q$ *) having input* $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ *such that columns of* $\mathbf{B}$ *spans* $\mathbb{Z}_q^n$ *and returns* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *& the trapdoor* $\mathbf{T} \in \mathbb{Z}^{m\times m}$ *such that*

- $\mathbf{AB}^T = \mathbf{0} \mod q$*, as well as distribution on* $\mathbf{A}$ *is statistically close to uniform.*
- *Moreover,* $\mathbf{T}$ *forms a basis of the lattice* $\mathcal{L}^{\perp}(\mathbf{A})$*, and its columns are distributed according to* $D_{\mathcal{L}^{\perp}(\mathbf{A}),s}$*, where* $s = C\sqrt{n\log q}\omega(\sqrt{\log m})$.

4.1.1. NIWI Proof for Lattice Problems

This signature also uses non-interactive witness indistinguishable (NIWI) proof for gap-language $L_{s,\gamma} = \{L_{Yes}, L_{No}\}$ where

$$L_{Yes} = \left\{ \begin{pmatrix} \mathbf{B}' \\ \mathbf{x}' \end{pmatrix} \middle| \exists \mathbf{y} \in \mathbb{Z}_q^n, i \in [N] : \|\mathbf{x}_i - \mathbf{B}_i^T\mathbf{y}\| \leq s\sqrt{m} \right\}$$

$$L_{No} = \left\{ \begin{pmatrix} \mathbf{B}' \\ \mathbf{x}' \end{pmatrix} \middle| \forall \mathbf{y} \in \mathbb{Z}_q^n, i \in [N] : \|\mathbf{x}_i - \mathbf{B}_i^T\mathbf{y}\| \geq \gamma.s\sqrt{m} \right\}$$

where $\begin{pmatrix} \mathbf{B}' \\ \mathbf{x}' \end{pmatrix} = \begin{pmatrix} \mathbf{B}_1 & \dots & \mathbf{B}_N \\ \mathbf{x}_1 & \dots & \mathbf{x}_N \end{pmatrix}$
and the language $L'_\gamma = \{L'_{Yes}, L'_{No}\}$, where

$$L'_{Yes} = \{(\mathbf{B}, \mathbf{x}, t) | \exists y : \|\mathbf{x} - \mathbf{B}^T\mathbf{y}\| \leq t\}$$
$$L'_{No} = \{(\mathbf{B}, \mathbf{x}, t) | \forall y : \|\mathbf{x} - \mathbf{B}^T\mathbf{y}\| \geq \gamma.t\}$$

$L_{s,\gamma}$ can be described using OR operation on several instances of L'_γ as follows

$$\begin{pmatrix} \mathbf{B}_1 & \dots & \mathbf{B}_N \\ \mathbf{x}_1 & \dots & \mathbf{x}_N \end{pmatrix} \in L_{Yes} \Leftrightarrow \bigvee\nolimits_i ((\mathbf{B}_i, \mathbf{x}_i, s\sqrt{m}) \in L'_{Yes}).$$

$$\begin{pmatrix} \mathbf{B}_1 & \dots & \mathbf{B}_N \\ \mathbf{x}_1 & \dots & \mathbf{x}_N \end{pmatrix} \in L_{No} \Leftrightarrow \bigwedge\nolimits_i ((\mathbf{B}_i, \mathbf{x}_i, s\sqrt{m}) \in L'_{No}).$$

It uses the method defined by Cramer et al. [30] to get the NIWI proof for $L_{s,\gamma}$ (negligible soundness error). It is non-interactive due to the transformations of Fiat-Shamir [35] in the random oracle. These results are summarized in the following lemma.

Lemma 8. *[37] There is a NIWI proof system for the language* $L_{s,\gamma}$ *, for* $\gamma \geq O(\sqrt{m/\log m})$ *in the random oracle model, where the length of the proof is* $O(mnN\log q)$ *bits.*

4.1.2. Signature Scheme

For the signature scheme, the scheme parameters are $q = poly(n), m \geq 8n \log q$, and $s \geq C\sqrt{n \log q}$ $\omega(\sqrt{\log m})$, where n is the security parameter. The hash function $H : \{0,1\}^\star \longrightarrow \mathbb{Z}_q^n$ is used as a random oracle.

- $G\_KeyGen(1^n, 1^N)$: Using $TrapGen(1^n, 1^m, q)$ generate $(\mathbf{B}_i, \mathbf{S}_i)_{i=1}^N$. Then generate $(\mathbf{A}_i, \mathbf{T}_i)_{i=1}^N \leftarrow OrthoSamp(1^n, 1^m, q, \mathbf{B}_i)$. Output the keys $pk = (\mathbf{A}_i, \mathbf{B}_i)_{i=1}^N$, $sk[i] = (\mathbf{T}_i)_{i=1}^N$ and the tracing key $tk = (\mathbf{S}_i)_{i=1}^N$
- $G\_Sign(sk[j], m)$: To obtain signature on message M with secret key T_j, first choose $r \leftarrow \{0,1\}^n$ and compute $\mathbf{y}_i = H(M\|r\|i)$, $i \in [N]$. Then,
  (i) generate a sample $\mathbf{e}_j \leftarrow SampPre(\mathbf{A}_j, \mathbf{T}_j, \mathbf{y}_j, s)$.
  (ii) for $i \neq j$ choose $\mathbf{e}_i \in \mathbb{Z}_q^m$ uniformly such that $\mathbf{A}_i\mathbf{e}_i = \mathbf{y}_i \mod q$.
  (iii) compute $\mathbf{z}_i = \mathbf{B}_i^T\mathbf{x}_i + \mathbf{y}_i$, where $\mathbf{x}_i$'s are sampled from $\mathbb{Z}_q^n$.

  Finally, construct a NIWI proof π for gap language $L_{s,\gamma}$ with witness using $(\mathbf{x}_i, i)$ and return the signature $\sigma = (r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$.
- $G\_Verify(pk, \sigma, M)$: Return 1 iff π is correct and $\mathbf{A}_i\mathbf{z}_i = H(M\|r\|i) \mod q$ for all $i \in [N]$.
- $G\_Open(tk, M, \Sigma)$: It output the smallest index i such that $dist(\mathcal{L}(\mathbf{B}_i^T), \mathbf{z}_i) \leq s\sqrt{m}$.

4.1.3. Secuirty Analysis

From the construction of $\mathbf{z}_i$'s, it is clear that the hardness of this signature scheme is based on *learning with error* problem. The following theorem guarantees the anonymity and traceability of the signature scheme based on LWE.

Theorem 5. *[37] The group signature scheme described above is anonymous if the proof used is witness indistinguishable and the $LWE_{m,q,\alpha}$ is hard for $\alpha = s/(q\sqrt{2})$, where m, n, s are according to the above scheme. Moreover, it is traceable if the $GapSVP_\gamma$, $\gamma = O(n \log^4 n)$ is hard.*

It was proved that if there is a successful forger who can forge the signature, then there exists another forger who is using the valid forgery returned by later can successfully solve the learning with error problem.

4.1.4. Remark

This first lattice-based construction of group signature gave a new road in the lattice-based signature, which leads to the more efficient group signature schemes [45, 51, 53, 68] which focused the linear signature size in terms of group size and worked on the drawback of the above scheme. Scheme [45] gave the first group signature, which has the signature and the public keys of the logarithmic size of the group. Along with the shorter key size, [51] signature gave the simpler construction and weaker security assumption. They also gave the ring setting for the above based on ideal lattices, which results in $\tilde{O}(\kappa \log N)(\kappa$ is security parameter, N is group size) bit size of the signature and public key both. Nguyen et al. [68] improved the above signature scheme by a factor of $O(\log N)$ in both signatures as well as public key using the non-interactive zero-knowledge proof with the security assumption based on SIS and LWE. Recently Ling et al. [53] gave the group signature, which produces the signature of fixed size that dependents only on the security parameter and independent of the group size. Group signature with additional features were also introduced in [22, 46, 49, 50, 52, 54]. Camenisch et al. [22] gave a generalization of group signature known as anonymous attribute tokens, where members sign anonymously with the help of credentials issued to them containing their attributes. They also generate tokens along with the signature, which contains a subset of their attributes. This paper introduced two schemes, one with the tracing property and another without the tracing property of the group manager. Langlois et al. [46] gave the first verifier-local revocation group signature scheme featuring logarithmic signature, membership revocation, and efficient hardness assumption. Libert et al. [49, 50] gave two group signature schemes first one is the first lattice-based group signature, which supports the dynamically increasing members of the group so that any new member can join the group at any time. The second one [50], it combines the group signature scheme [51] with two layers of identity-based encryption and zero-knowledge proof to introduce the first lattice-based group signature with the message-dependent opening. Message dependent opening scheme empowers the signer so that his identity can only if revealed if a special authority called admitter reveals the trapdoor for the corresponding message that the signer has signed. All the above schemes exploited the GPV trapdoor [38], but Libert et al. [48] gave the first lattice-based group signature that completely ignores the GPV trapdoor with efficient parameter choice. Table 2 states the signature size, group public key size, and secret signing key size of each scheme based on security parameter κ and maximum expected size of the group $N = 2^l$.

Sig. Scheme	Secret key size	Public key Size	Sign. Size
[37]	$\tilde{O}(\kappa^2)$	$\tilde{O}(\kappa^2.N)$	$\tilde{O}(\kappa^2.N)$
[22]	$\tilde{O}(\kappa^2)$	$\tilde{O}(\kappa^2)$	$\tilde{O}(\kappa^2.N)$
[45]	$\tilde{O}(\kappa^2)$	$\tilde{O}(\kappa^2.l)$	$\tilde{O}(\kappa.l)$
[46]	$\tilde{O}(\kappa.l)$	$\tilde{O}(\kappa^2.l)$	$\tilde{O}(\kappa.l)$
[68]	$\tilde{O}(\kappa^2)$	$\tilde{O}(\kappa^2.l^2)$	$\tilde{O}(\kappa + l^2)$
[51]	$\tilde{O}(\kappa)$	$\tilde{O}(\kappa^2.l)$	$\tilde{O}(\kappa.l)$
[48]	$\tilde{O}(\kappa.l)$	$\tilde{O}(\kappa^2 + \kappa.l)$	$\tilde{O}(\kappa.l)$
[49]	$\tilde{O}(\kappa)$	$\tilde{O}(\kappa^2.l)$	$\tilde{O}(\kappa.l)$
[50]	$\tilde{O}(\kappa)$	$\tilde{O}(\kappa^2.l)$	$\tilde{O}(\kappa.l)$
[52]	$\tilde{O}(\kappa) + l$	$\tilde{O}(\kappa^2 + \kappa.l)$	$\tilde{O}(\kappa.l)$
[53]	$\tilde{O}(\kappa)$	$\tilde{O}(\kappa)$	$\tilde{O}(\kappa)$
[54]	$\tilde{O}(\kappa) + l$	$\tilde{O}(\kappa^2 + \kappa.l)$	$\tilde{O}(\kappa.l)$

Table 2: Comparison of Group Signature Schemes

The above signatures have theoretical advances but can't be used for practical application. Thus the research direction involves developing a more efficient group signature that is suitable for practical implementation. There exist many classical variants of group signature schemes such as proxy group signature, group blind signature, group designated verifier signature scheme, and many more. These signature schemes were introduced as they are more practically suitable for various applications. Thus, developing a quantum resistance variant of these schemes is still open for the research community.

4.2. Ring Signature Scheme

Ring signature scheme [77] can be viewed as a group signature scheme, but without any group leader, hence a signer can't be traced. Due to this facility, this scheme was first introduced as a way to leak secrets. Wang et al. [84] gave the first ring signature over lattices in the random oracle model using lattice basis delegation methods [23, 24, 69]. In [84] they also described their scheme in the standard model. Before giving the signature scheme, we describe the *GenSampPre* algorithm for sampling preimage in the extended lattice.

This algorithm was first introduced in [23], but with different parameters and different structure of the extended lattice. Wang et al. improved the algorithm according to their signature scheme.

4.2.1. Pre-Image Sampling for Extended Lattice

Let us denote the set of all integer by $\mathbb{Z}_{>0}$. For some $w, w_1, w_2, w_3, w_4 \in \mathbb{Z}_{>0}$ with $w = w_1 + w_2 + w_3 + w_4$, let $K = [w]$ and we express $\mathbf{A}_K = [\mathbf{A}_{K_1}, \mathbf{A}_{K_2}, \mathbf{A}_{K_3}, \mathbf{A}_{K_4}] \in \mathbb{Z}_q^{n \times wm}$, where $\mathbf{A}_{K_i} \in \mathbb{Z}_q^{n \times w_i m}$, $i \in [4]$. Consider $\mathbf{A}_S = [\mathbf{A}_{w_1} \| \mathbf{A}_{w_3}] \in \mathbb{Z}_q^{n \times (w_1+w_3)m}$ along with short basis $\mathbf{B}_S$ of the lattice $\mathcal{L}^{\perp}(\mathbf{A}_S)$ and given an integer $r \geq \|\widetilde{\mathbf{B}}_S\| \omega(\sqrt{\log n})$, the algorithm *GenSampPre* returns a preimage of the function $f_{\mathbf{A}_K}(\mathbf{e}) = \mathbf{A}_K \mathbf{e} \mod q$. $GenSampPre(\mathbf{A}_\mathbf{K}, \mathbf{A}_S, \mathbf{B}_S, \mathbf{y}, r)$ runs as follows :

1. Using distributions $D_{\mathbb{Z}^{w_2 m}, r}$ and $D_{\mathbb{Z}^{w_4 m}, r}$, sample $\mathbf{e}_{K_2} \in \mathbb{Z}^{w_2 m}$ and $\mathbf{e}_{K_4} \in \mathbb{Z}^{w_4 m}$. Express $\mathbf{e}_{K_2}$ as $[\mathbf{e}_{w_1+1}, \ldots, \mathbf{e}_{w_1+w_2}] \in \mathbb{Z}^{w_2 m}$ and $\mathbf{e}_{K_4}$ as $[\mathbf{e}_{w-w_4+1}, \ldots, \mathbf{e}_w] \in \mathbb{Z}^{w_4 m}$.
2. Define $\mathbf{z} = \mathbf{y} - \mathbf{A}_{K_2}\mathbf{e}_{K_2} - \mathbf{A}_{K_4}\mathbf{e}_{K_4} \mod q$. Run $SamplePre(\mathbf{A}_S, \mathbf{B}_S, \mathbf{z}, r)$ (from [38]) to sample a vector $\mathbf{e}_S \in \mathbb{Z}^{(w_1+w_3)m}$ from the distribution $D_{\mathcal{L}^{\perp}_{\mathbf{y}}(\mathbf{A}_\mathbf{K}), r}$. Write $\mathbf{e}_S = [\mathbf{e}_1, \ldots, \mathbf{e}_{w_1}, \mathbf{e}_{w_1+w_2+1}, \ldots, \mathbf{e}_{w-w_4}] \in \mathbb{Z}^{(w_1+w_3)m}$ and let $\mathbf{e}_{K_1} = [\mathbf{e}_1, \ldots, \mathbf{e}_{w_1}] \in \mathbb{Z}^{w_1 m}$, $\mathbf{e}_{K_3} = [\mathbf{e}_{w_1+w_2+1}, \ldots, \mathbf{e}_{w-w_4}] \in \mathbb{Z}^{w_3 m}$.
3. Output $\mathbf{e} = \{\mathbf{e}_{K_1}, \mathbf{e}_{K_2}, \mathbf{e}_{K_3}, \mathbf{e}_{K_4}\} = [\mathbf{e}_1, \ldots \mathbf{e}_w] \in \mathbb{Z}^{wm}$.

Note 8. *According to construction, we have* $\mathbf{A}_{K_1}\mathbf{e}_{K_1} + \mathbf{A}_{K_3}\mathbf{e}_{K_3} = \mathbf{A}_S\mathbf{e}_S = \mathbf{z} \mod q$. *Therefore,* $\mathbf{A}_K\mathbf{e} = \Sigma_{i=1}^4 \mathbf{A}_{K_i}\mathbf{e}_{K_i} = \mathbf{y} \mod q$, *and the output* $\mathbf{e}$ *is contained in* $\mathcal{L}^{\perp}_{\mathbf{y}}(\mathbf{A}_K)$. *From Theorem 3.4 in [23] ,* $\mathbf{e}$ *is within a negligible statical distance of* $D_{\mathcal{L}_{\mathbf{y}}{}^{\perp}(\mathbf{A}_K), r}$.

4.2.2. Ring Signature Scheme in Random Oracle

This section describes the Wang and Sun's [84] signature scheme. Suppose l, m, n, q, t are positive integers such that $q \geq 2$ and $m \geq 5n \log q$. Moreover, $\tilde{L}, r$ are the other system parameters defined as follows :

- $\tilde{L} \geq O(\sqrt{n \log q})$, an upper bound on the Gram-Schmidt size of a participant's secret basis.

- $r \geq \overline{L}\omega(\sqrt{\log q})$ is the Gaussian parameter used to generate the required secret basis and short vectors.

The scheme uses a hash function $H_1 : \{0,1\}^\star \longrightarrow \mathbb{Z}_q^n$ which is modelled as a random oracle.

1. $R.KeyGen(\lambda)$: Generates $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$ with a basis $\mathbf{B}_i \in \mathbb{Z}^{m\times m}$ for lattice $\mathcal{L}^\perp(\mathbf{A}_i)$ for each ring member using $TrapGen(1^\lambda)$ [38], where $\|\mathbf{B_i}\| \leq \overline{\mathbf{L}}$ (Theorem 3.2 [69]). $KeyGen$ returns the keys $(pk_i = \mathbf{A}_i, sk_i = \mathbf{B}_i)$ for each member i . Define $R = \{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$ for simplicity

2. $R.Sign(R, sk_i, M)$: For a ring of N members with public keys, define $\mathbf{A}_R = [\mathbf{A}_1\| \ldots \|\mathbf{A}_N] \in \mathbb{Z}_q^{n\times Nm}$, for $i \in [N]$. To generate the signature on message $M \in \{0,1\}^\star$, member i proceeds as follows :
   - Computes $\mathbf{y} = H_1(M) \in \mathbb{Z}_q^n$ and defines a label lab_R that describes how $\mathbf{A}_R$ is associated with ring members $\{1, \ldots, N\}$.
   - Generate $\mathbf{e} \leftarrow GenSampPre(\mathbf{A}_R, \mathbf{A}_i, \mathbf{B}_i, \mathbf{y}, r) \in \mathbb{Z}^{Nm}$ distributed according to $D_{\mathcal{L}_\mathbf{y}^\perp(\mathbf{A}_R),r}$.
   - $R.Sign$ Returns the signature $\sigma = (e, lab_R)$.

3. $R.Verify(R, M, \sigma)$: This algorithm returns 1 if $0 \leq \|\mathbf{e}\| \leq r\sqrt{Nm}$ and $\mathbf{A}_R\mathbf{e} \mod q = H_1(M)$ else returns 0.

The scheme is proved fully anonymous and unforgeable with regards to collision-resistant of the hash function and hardness of SIS problem, described in the theorem (Theorem 1,2 [84]) below.

Theorem 6. *[84] The signature scheme described above is fully anonymous under the hardness of $ISIS_{q,NM,r}$. Moreover, if the hash function H is collision-resistant, and $SIS_{q,Nm,2r}$ is hard then the scheme is unforgeable also.*

4.2.3. Ring Signature Scheme in Standard Model

This scheme [84] is motivated by the scheme proposed by Boyen's [18] who gave the proper framework for the signature scheme in the standard model. The scheme works as follows :

- $R.KeyGen(1^\lambda)$: Same as the above scheme.
- $R.Sign(R, sk_i, M)$: Given the public keys $R = \{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$, the private keys sk_i and the message $M \in \{0\} \times \{0,1\}^d$. To generate the signature, member i proceeds as follows :
  (i) Computes $C_M = \Sigma_{i=0}^d (-1)^{M[i]} C_i \in \mathbb{Z}_q^{n\times m}$, where $C_i \in \mathbb{Z}_q^{n\times m}$, , $i = 0, \ldots d$ are chosen as public parameter.
  (ii) Defines $\mathbf{A}_R = [\mathbf{A}_1\| \ldots \|\mathbf{A}_N\|C_M] \in \mathbb{Z}_q^{n\times(N+1)m}$ and sets a label lab_R that describes how $\mathbf{A}_R$ is associated with ring members $\{1, \ldots, N\}$.
  (iii) Generate $\mathbf{e} \leftarrow GenSampPre(\mathbf{A}_R, \mathbf{A}_i, \mathbf{B}_i, \mathbf{y}, r) \in \mathbb{Z}^{Nm}$ distributed according to $D_{\mathcal{L}_\mathbf{y}^\perp(\mathbf{A}_R),r}$.
  (iv) $R.Sign$ Returns the signature $\sigma = (e, lab_R)$.
- $R.Verify$ This algorithm returns 1 if $0 \leq \|\mathbf{e}\| \leq r\sqrt{(N+1)m}$ and $[\mathbf{A}_1\| \ldots \|\mathbf{A}_N\|\Sigma_{i=0}^d (-1)^{M[i]} C_i]e = 0 \mod q$ else it returns 0.

4.2.4. Security Analysis

The theorem (Theorem 3 [84]) stated below proves the unforgeability of the above scheme under the hardness of SIS.

Theorem 7. *[84] The above described schemes are unforgeable if the $SIS_{q,Nm,r}$ is hard.*

Security analysis of both the ring signatures proves that there exists a constructor who utilizes the valid forgery returned by a successful forger and either finds a collision for the hash function or solves the SIS problem for the later one. For ring signature in the standard model, we can reduce the security to the hardness of the SIS problem.

4.2.5. Remark

Lattice-based ring signature came into the limelight through the work of Brakerski and Kalai in 2010 [19]. They constructed the ring trapdoor function utilizing the SIS problem. They gave the foundation ring signature in the standard model. At the same time, Wang et al. [83] also developed the ring signature in the standard model exploiting the features of bonsai trees. In 2011 Wang and Sun [84] gave two signature schemes that we discussed above in standard as well as in a random oracle. Aguilar-Melchor et al. [4] developed the ring signature using the model developed by Lyubachevsky [57] based on SVP. This scheme has a smaller secret and public keys and is anonymous, even full key exposure. It was the first ring signature based on LWE problem yielding signatures in linear size. Libert et al. [48] gave the ring signature with logarithmic signature size $\tilde{O}(logN.n)$ of the ring size based on the Merkle-tree construction. Lyubashevsky's [58] lattice signature without trapdoor was also extended to develop a new ring signature [86], which provides a more efficient and shorter signature.

The hardness of this scheme was reduced to the hardness of the SIS problem using rejection sampling. Wen et al. [87] gave a ring signature in the standard model based on the split-SIS problem; moreover, in this signature public key of the ring doesn't increase with the members of the ring. Table 3 compares each signature scheme for the secret key, public key, and signature size.

Sig. Scheme	Secret key size	Public Key Size	Sign Size
[83]			$(l+d+1)m$
[84](ROM)	Nm^2	$Nmnlogq$	$lnmlogq + lm$
[84](SM)	Nm^2	$(Nmn+d+1)logq$	$lnmlogq + lm$
[4]	$N(3+2c/3logn)$	$N(n-1)logp$	$l + l(3+2c/3logn) + (n-1)logp$
[86]	$Nmnlogq$	Nm^2	$lm + k$
[87]	$N.4m^2logq$	$(3mn+n+dn)logq$	$(p+1)mlogq$

Table 3: Comparison of Ring Signature Scheme

Signer anonymity, along with linkability, is very desirable for many applications such as e-voting, ad-hoc authentications, and many more. Today, linkable ring signatures provide a solution to protect the sender's privacy in cryptocurrency transactions. Torres et al. [82] proposed the first lattice-based linkable ring signature, which used the BLISS signature [58] to introduce linkability. In the same year, 2018, Baum et al. [12] also proposed a linkable signature scheme which claimed to have a shorter signature, but both the schemes don't say anything about their implementation. Moreover, both the schemes involve rejection sampling, and the signature size grows linearly with the number of users, which becomes an overhead in terms of performance. In 2019, Xingye et al. [90] introduced a practical linkable ring signature based on lattices. They proved the scheme to be as fast as classical systems and results in a much shorter signature. They introduced a new primitive called Chameleon hash plus, which was instantiated using NTRU lattice as well as standard SIS problem; thus, it doesn't depend on one-way trapdoor permutation. For security level 100 and with 2^{10} users, it gives a signature of $1301.9KB$, and the other two schemes give $9770KB$ and $9360KB$, clearly [90] proposed a better alternative along with practical implementation also. But in all the above schemes, signature size grows linearly with the number of users in the ring. Thus, developing a signature that doesn't increase with the number of users is still an open problem.

Ring signature without trapdoor generates the shorter signature, but the scheme exists in the random oracle model. Developing a signature scheme in the standard model with some new techniques to improve the signature size and to implement them in blockchains is quite an exciting field of research. Lattice-based ring signature variants like traceable ring signature, verifiable ring signature forms promising candidates for future construction.

4.3. Proxy Signature Scheme over lattices

Mambo introduced the idea of proxy signature in 1996 [61]. In this scheme, the original signer delegates his/her signing rights to the proxy signer. There are three types of delegations, namely, full delegation, partial delegation, and delegation by warrant defined as follows :

1. *Full Delegation*: In such delegation, the original signer and proxy signer share the same secret key. Thus signature from the proxy signer is indistinguishable from the original signer's signature.

2. *Partial Delegation*: Here, the original signer generates a new secret key for proxy signer from the secret. Such delegation results in two signature schemes :

   (i) *Proxy-Unprotected Signature*: Here, both original and proxy signers can generate a valid proxy signature; only a third party can't create a valid signature.

   (ii) *Proxy-Protected Signature*: Here, only proxy signer can create a valid signature. Neither the original signer nor any third party can generate a valid signature.

3. *Delegation by Warrant*: Original signer delegates signing rights for an interval time using a warrant. This delegation has two approaches.

(i) *Delegate Proxy*: In this approach, the original signer signs a document declaring the proxy signer under any signing scheme.

(ii) *Bearer Proxy*: Here, the warrant consists of a message part and the original signer's signature on the newly generated proxy keys.

A Proxy Signature scheme should satisfy the following properties [61] :

- Distinguishability: Signature generated by proxy signer should be distinguishable from the signature from the original signer.
- Unforgeability: Any third party, other than designated proxy signer, should not be able to generate a proxy signature.
- Verifiability: Proxy signer should be able to prove the consent of the original signer on the signed message.
- Identifiability: The original signer can determine the proxy signer's identity from the proxy signature.
- Non-Repudiation: A proxy signer cannot disavow an accepted proxy signature generated by him.

Now, we describe the proxy signature introduced by Wang et al. [29], which is quantum computer resistant. It was developed using the bonsai tree [24] and the preimage sampling [38]. For delegating signing right, Wang used the bonsai tree principal and its variety [2].

4.3.1. Signature scheme

The scheme consist of following parameters. Let n be a prime, $m \geq 2n \log q$, and $q \geq \beta\omega(\log n); \beta = poly(n)$. Along with other system parameter, bound $\tilde{L} \geq O(\sqrt{n \log q})$ and the Gaussian parameter $r = \tilde{L}\sqrt{\log n}$. Moreover, it uses two hash functions defined as $H_1 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^n$ and $H_2 : \mathbb{Z}_q^\star \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^n$.

1. *P.KeyGen* : Using $TrapGen$ original signer generates $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A} \in \mathbb{Z}_q^{m\times m}$ of the lattice $\mathcal{L}^\perp(\mathbf{A})$. Similarly, proxy signer P_i generates a random matrix $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_{A}}_i \in \mathbb{Z}_q^{m\times m}$. Two random matrices $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{K} \in \mathbb{Z}_2^{m\times n}$ are set as public keys by the original signer. It outputs the public keys and secret keys of original signer and the proxy signers as $(\mathbf{A}, \mathbf{B}, \mathbf{K})$, $\mathbf{A}_i$, and $\mathbf{T_A}$, $\mathbf{T_{A}}_i$ respectively.
2. *P.SignDel*: For delegating signing rights to the proxy signer P_i, original signer proceeds as follows:
   - Computes a warrant w_i for the proxy signer P_i, consisting of delegation period, the identities, public keys of the original signer and the proxy signer, and the signature of the original signer, using his secret key with the help of signature scheme by Gentry et al. [38].
   - Next, he computes $\mathbf{KA}_i \mod 2$, and computes $\mathbf{R}$ by replacing all 0's with -1's in $\mathbf{KA}_i \mod 2$.
   - Then, using the Bonsai trees algorithm [24] he generates a short basis $\mathbf{T}' \in \mathbb{Z}_q^{2m\times 2n}$ of the lattice $\mathcal{L}^\perp(\mathbf{A}, (\mathbf{AR} + \mathbf{B}))$
   - Finally, he sends $\mathbf{T}'$ and the warrant w_i to the proxy signer secretly.

   Proxy Signer P_i checks the warrant w_i and accepts the proxy key iff $(\mathbf{A}, (\mathbf{AR}+\mathbf{B}))\mathbf{T}' = 0 \mod q$ and $\|\mathbf{T}'\| \leq s\sqrt{2m}$.
3. *P.Sign*: Proxy signer proceeds as follows to generate the signature on the message M :
   - He computes $h_1 = H_1(M, r)$, for some random $r \in \{0,1\}^l$ and $\mathbf{KA}_i \mod 2$. Then computes $\mathbf{R}$ by replacing all 0's with -1 in $\mathbf{KA}_i \mod 2$.
   - Generates $\mathbf{e}_1 \leftarrow SampPre((\mathbf{A}, (\mathbf{AR} + \mathbf{B})), h_1, s)$
   - Next, he computes $h_2 = H_2(w_i, r)$ and $\mathbf{e}_2 \leftarrow SampPre(\mathbf{A}_i, h_2, s)$.

     Then returns the signature $\sigma = (\mathbf{e}_1, \mathbf{e}_2, r, w_i)$.
4. *P.Verify*: He first computes $\mathbf{R}$ by replacing all 0's with -1's in $\mathbf{KA}_i \mod 2$, from the public keys. Then, he checks $(\mathbf{A}, (\mathbf{AR} + \mathbf{B}))\mathbf{e}_1 = 0 \mod q$, $\mathbf{A}_i\mathbf{e}_2 = 0 \mod q$ and $\|\mathbf{e}_1\| \leq s\sqrt{2m}$

4.3.2. Security Analysis

For the unforgeability of signature scheme, we consider for three types of adversaries, type 1 (original signer tries to impersonate as a proxy signer) having the knowledge of secret key of the original signer, type 2 (any third party tries to impersonate as a proxy signer) having the knowledge of the secret of the proxy signer and type 3 (any third party) adversary having no additional information except all the public information.

The security analysis proves unforgeability under the hardness of the SIS problem for type 1 and type 2 adversaries by assuming the existence of a successful adversary and using the valid forgery for solving the SIS problem. Thus, the security reduces to the hardness of hard SIS problem of lattices. For type 3, it is clearly derived as a conclusion from the latter two. Other security assumptions are proved heuristically assuming that hardness of lattice problems

4.3.3. Remark

In 2010, Jiang et al. [40] introduced the first lattice-based proxy signature, but this scheme was proved to be insecure by Tian and Huang [80], where they showed that anyone could forge a proxy signature on any message. Cash et al. [24] introduced a new insight of bonsai trees in the proxy signature. Since then, many proxy signatures were developed, among which the signature given by Wang et al. [29] in 2011, as we discussed above, is one of them. Xia et al. [88] also gave proxy signature using the bonsai trees, which security is based on the hardness of average-case SIS and ISIS problem. This scheme was also proved existentially unforgeable under the chosen-message attack in the random oracle. But both the signature schemes face a drawback that they produce longer signatures, which is inefficient for practical use. Using fixed dimension basis expansion techniques [2], Yu Lei [47] in 2013 developed more efficient proxy signature. But this signature uses the basis expansion along with two times a pre-image sampling algorithm, which turns out to be time-consuming and not efficient for practical uses. Yang et al. [92] in 2015 developed proxy signature without trapdoor, extending the concept of signature developed by Lyubashevsky [58] which generated shorter signature in comparison to above schemes and proved the scheme to be existentially unforgeable in random oracle.

Proxy signatures with additional features were also proposed by distinguish cryptographers. The lattice-based identity-based proxy signature was first proposed by Zhang et al. [94] using the bonsai tree, which is proved to have unforgeability proxy key, revocability of proxy signature as well as existential unforgeable. But this signature scheme was not proxy protected. Kim et al. [42] gave the first identity-based proxy protected signature scheme in the random oracle model. Identity-based proxy signature in the standard model [85] was also developed utilizing the lattice-based delegation techniques [2] and lattice-based signing. Then Zhang et al. [96] also used the concept of blind signature and gave the identity-based proxy blind signature, but Rawal et al. [73] carried an attack over this scheme exposing the secret master key of the scheme. Lattice-based multigrade proxy signature [95] was also proposed by Zhang et al. in 2013 based on SIS and ISIS hardness assumptions. Table 4 summarizes the above discussion by comparing the schemes based on their key size and signature size.

Existing proxy signature schemes that give shorter signatures are based on random oracle; therefore, future direction involves eliminating the random oracle model and improving the signature size. It would be interesting to construct other lattice-based proxy signature schemes such as multi-proxy, proxy-multi, multi-proxy-multi signature schemes, etc.

4.4. Blind Signature

David Chaum introduced the concept of blind signature [27] in 1983, which became a cornerstone in the area of cryptography. Markus Ruckert [76] gave the first lattice-based blind signature scheme in 2010. This scheme is motivated by the Lyubashevsky scheme [56] along with the Fiat-Shamir paradigm [35]. It involves operations having quasi-linear complexity concerning the main parameter n, all keys and signatures require a quasi-linear amount of storage bits.

This signature scheme satisfies the two properties *blindness* and *One-more unforgeable* [14, 21].

4.4.1. Signature Scheme

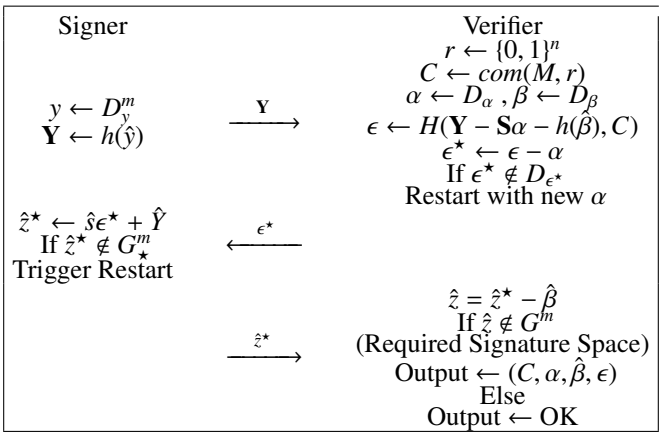
This section describes the signature schemes consisting of a three tuple $(B.KeyGen, B.Sign, B.Verify)$ which works as follows :

- $B.KeyGen(1^n)$: Pick a secret key $\hat{s} \leftarrow D_s^m$, and let $h \leftarrow \mathcal{H}(\mathbf{R}, M)$ be a compression function. Let $C(1^n) : \{0,1\}^\star \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a commitment scheme. Pick a function $com \leftarrow C(1^n)$ and $H \leftarrow \mathcal{H}(1^n)$, which maps $\{0,1\}^\star \leftarrow D_\epsilon \subset D$. Then, return the public key $\mathbf{S} \leftarrow h(\hat{s})$.

- $B.Sign$: To generate a signature on a message $M \in \{0,1\}^\star$ this algorithm proceeds as follows as shown in the following table:

Scheme	Secret key size		Public key Size		Sign. Size
[40]	O. Signer	P. Signer			
[29]	O. Signer $m^2 logq$	P. Signer $5m^2 logq$	O. Signer $2mnlogq + mn$	P. Signer $mnlogq$	$3mlogq$ $+l$
[88]	O. Signer $m^2 logq$	P. Signer $8m^2 logq$	O. Signer $mnlogq$	P. Signer $mnlogq$	$2mlogq$
[47]	O. Signer $m^2 logq$	P. Signer $2m^2 logq$	O. Signer $mnlogq$	P. Signer $2mnlogq$	$2mlogq$
[92]	O. Signer $(mklogq)$ $*(2d+1)$	P. Signer $(mklogq)$ $*(2d+1) + mlog(12\sigma)$	O. Signer $nklogq$	P. Signer $nklogq$	$2mlog(12\sigma)$
[42]	O. Signer $2m^2 logq$	P. Signer $m^2 logq$	O. Signer $mnlogq$	P. Signer $mnlogq$	$2mlogq$
[85]	O. Signer $2m^2 logq$	P. Signer $m^2 logq$	O. Signer $mnlogq$	P. Signer $mnlogq$	$4mlogq$
[95]	O. Signer $m^2 logq$	P. Signer $2m^2 logq$	O. Signer $mnlogq + m + d$ (d : msg length)	P. Signer $2mnlogq$	$2m$

Table 4: Comparison of Proxy Signature Scheme



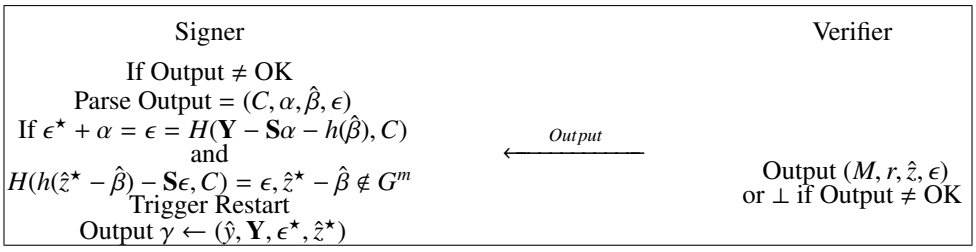


Table 5: Blind Signature Scheme

Thus, the signing algorithm returns the signature $(r, \hat{z}^{\star}, \epsilon)$ on the given message M. Here, the thing to be noted, when restart is encountered, then we pick new α, and when trigger restart is encountered, then we run the algorithm again with new r to make the new run independent of the previous one. But, there is an exception at last in step 5; here, if the verifier triggers restart, then the signer can stop the algorithm as there is a chance that the verifier has a valid signature call for a restart.

- $B.Verify$: This algorithm returns 1 if $\hat{z} \in G^m$ and $H(h(\hat{z}) - \mathbf{S}\epsilon, Com(M, r)) = \epsilon$, else return 0.

4.4.2. Security Analysis

The Above signature scheme has been proved to satisfy the blindness, one-more unforgeable, and leakage resilient; we can refer [76] for the proof of these properties. This paper gives rigorous prove of all the above properties under the hardness of the shortest integer solution problem.

4.4.3. Remark

The signature that we discussed above was the first step towards the blind signature scheme in lattice cryptography. But this scheme also suffers drawbacks; it needs to restart several times to get a valid signature, which results in failing to generate a signature for a certain probability. Then, a two-move blind signature using preimage sampling was introduced by Wang et al. [39], which doesn't have any signing procedure drawbacks. Identity-based blind signatures were also introduced by Gao et al. [36], which was proved to be unconditionally blind in the standard model. The above schemes were unconditionally blind. Tian et al. [81] gave a partially blind signature scheme in 2016 using Lyubashevsky's signature scheme [58] and Abe and Okamoto's construction [1] of partially blind signature in random oracle. Recently, Quoc et al. [71] proposed a quantum-resistant signature scheme using Dilithium [34], a promising candidate submitted for standardization to NIST under the hardness of Module LWE and Module SIS problem, which results in shorter signatures.

These were few schemes that were developed recently, but there are many stones that need to be unturned in the quantum-resistant blind signature scheme as this signature scheme has many applications. Thus, developing a blind signature scheme that is easy to implement practically is still a research gap.

Here, future direction involves constructing signature schemes with less interaction and a reduced number of repetitions or restarts between the signature generation and trying lattice-based fair blind, restrictive blind, restrictive partially blind signatures with efficient structure and signature size.

4.5. Threshold Signature Scheme

In the above sections, we discussed two concepts, group signature and ring signature, which allows a member to sign anonymously on behalf of the other members. Bresson, Stern, and Szydlo [20] gave a variation of the ring signature where out of N members at least t can jointly sign the document without compromising their anonymity. Concerning the lattice-based threshold signature, Cayrel et al. [26] gave the first signature in this path using Aguilar's [3] and Cayrel's [25] results. This scheme generates small signatures in less number of executions rounds, and its security is based on the hardness of the SIS problem. The scheme is proved to be existentially unforgeable, and the source is hiding. We first discuss what is meant by source hiding.

Definition 22. ***Source Hiding :*** *A threshold signature scheme is said to be source hiding if the same signature can be produced on a given message with different subsets of signers from the N signers.*

4.5.1. Signature Scheme

The *Setup* algorithm outputs the system parameters n, m, q on the input of the security parameter k.

- $T.KeyGen$: For N members we choose the public key $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ and the private key as $\mathbf{x}_i$ with the binary vectors with hamming weight $m/2 + 1$.

- $T.Sign$: This algorithm takes as input the message to be signed and the public keys of the N members and the private keys of the t members who want to sign. Out of the t signer, one acts as a leader L. The algorithm then proceeds as follows: each pair of $(Signer_i, L)$ execute the identification scheme (given below) where the $Signer_i$ is modeled as a prover and L as a verifier sharing the challenges α and b. $(L, Verifier)$ also runs the identification scheme where the commitments and answers are the compositions of the values gained by the leader from the signers. For non-signer, the leader uses the substitute private keys of null vectors. Then to maintain anonymity, the leader applies the block permutation to the individual values obtained and outputs the signature as the transcript of the interaction between the leader and the verifier.

- $T.Verify$: The signature of the scheme consists of the transcript of the sequence of a round of the identification scheme. Thus in verification, we check whether the commitment is correct for every round for the corresponding challenge. The signature is accepted if the check hold in every round; else, it is discarded.

Generalised CLRS Identification Scheme (GCLRS)

Procedure : Identification Scheme
$M' = \{Memebers\}$ with $S' = \{Signer\} \subset M'$ where $|S'| = t$, $|M'| = N$
• Prover(Pass 1):
First, for each member commitments are obtained as follows:
each signer $i \in S'$ computes
$\sigma_i \leftarrow S_{m+1}$, $\mathbf{u}_i \leftarrow \mathbb{Z}_q^{m+1}$, $\mathbf{r}_{0,i} \leftarrow \{0,1\}^n$, $\mathbf{r}_{1,i} \leftarrow \{0,1\}^n$
$c_{0,i} \leftarrow Com(\sigma_i \| \mathbf{A}_i\mathbf{u}_i, \mathbf{r}_{0,i})$, $c_{1,i} \leftarrow Com(\sigma_i(\mathbf{u}_i) \| \sigma(\mathbf{x}_i), \mathbf{r}_{1,i})$
and gives $c_{0,i}$, $c_{1,i}$ to L.
For non-signers $j \in M'/S'$, L computes the commitments with $\mathbf{x}_i = 0$.
Then, L chooses a random constant $n-$ block permutation on N blocks Σ to get the master commitments $C_0 = Com(\Sigma \| c_{0,1} \| \dots \| c_{0,N}, \mathbf{r}_0)$ and $C_1 = Com(\Sigma \| c_{1,1} \| \dots \| c_{1,N}, \mathbf{r}_1)$ and sends them to V.
• Verifier(Pass 2):
V sends $\alpha \in \mathbb{Z}_q^*$ to L, it is needed to verify the previous commitment and L passes it to S'.
• Prover (Pass 3) :
On receiving α each signer $i \in S'$ computes $\beta_i \leftarrow \sigma_i(\mathbf{u}_i - \alpha\mathbf{x}_i)$
for the non-signer $j \in M'/S'$, L computes β_j but with $\mathbf{x}_j \leftarrow 0$
Then, L sends $\beta = \Sigma(\beta_0, \dots, \beta_{N-1})$ to V
Challenge :
• Verifier (Pass 4) :
V sends $b \leftarrow \{0,1\}$ as a challenge to L, which is passed on to S' by L
Response :
• Prover (Pass 5) :
Signer responds to challenge as follows
each signer $i \in S'$
if $b = 0$ reveal σ_i and if $b = 1$ reveal $\sigma_i(\mathbf{x}_i)$ to L.
Then L sets $\sigma = (\sigma_0, \dots \sigma_{N-1})$, and responds to V as follows :
if $b = 0$ then L sends $\Pi = \Sigma \circ \sigma$ and $\Pi(\mathbf{r}_{0,0}, \dots, \mathbf{r}_{0,N-1})$
if $b = 1$, L sends $\Pi(x) = \Sigma(\sigma_1(x), \dots \sigma_{N-1}(x))$ and $\Pi(\mathbf{r}_{1,0}, \dots, \mathbf{r}_{1,N-1})$
• Now, V verifies the correctness of *master commitments*, permutation and hamming weight.
If $b = 0$ then V checks $c_0 = Com(\Sigma \| \mathbf{A}\Pi^{-1}(\beta) \| \mathbf{r}_0)$ or not and Π is well formed .
If $b = 1$ then V checks $c_1 = Com(\beta - \alpha\Pi(x) \| \Pi^{-1}(\beta) \| \mathbf{r}_1)$ or not and $\Pi(x)$ has hamming weight $t(m/2 + 1)$.

4.5.2. Security Analysis

The scheme has its hardness based on the SIS problem in the random oracle model and also proved to source hiding unconditionally. The theorem stated below [26] discusses the security analysis of the scheme.

Theorem 8. *[26] The scheme is an honest verifier zero-knowledge proof of knowledge, with soundness error less than* $1/2$ *, that t members who sign a known vector v of length* $N(m+1)$ *and hamming weight* $t(m/2+1)$ *such that each N block of size m either weights* $m/2 + 1$ *or zero. The scheme is secure in the random oracle model under the hardness of the SIS problem.*

4.5.3. Remark

In the scheme given by Cayrel et al. [26], each signer has its own public key; thus, the verification time grows linearly with the number of users or number of signers. Then the threshold signature scheme based on gaussian sampling by Bendlin et al. [15] and commitment and zero-knowledge protocols of Baum et al. [11] introduced compact threshold cryptosystem. But these can produce a prior number of online non-interactions signature/verification after an offline interactive step. Bendlin et al. [16] gave a threshold version of Regev's CPA-secure encryption scheme [75], and Myers et al. [66] applied the technique to fully homomorphic encryption. Xie et al. [89] gave a threshold scheme from lossy trapdoor functions, which can also be instantiated from LWE. All the above schemes produce signatures and public keys linear in size in N. Rawal et al. [72] proposed a threshold ring signature with message block sharing, which is both unforgeable and anonymous. Message block sharing seems efficient for fewer users, but if the number of users increases, this technique is not helpful.

Developing a lattice-based threshold signature without any trapdoor function is an essential area for future expansion as a signature scheme without trapdoor results in shorter signatures. One can try lattice linkable threshold, traceable threshold signature, threshold

proxy, and many signatures that have significant practical applications.

4.6. Attribute-based Signature

Maji et al. [60] introduced the Attribute-based signature scheme in 2011, where the signer satisfying a set of attributes can only obtain the signing rights and can sign the message with any predicates fulfilled by his attribute set. This signature scheme provides privacy protection, good expression ability, and many cryptographic applications such as anonymous authentication, access control, and attribute messaging.

After this outbreak of Maji et al. [60], many improvements on attribute signature were proposed, but all were based on discrete log problem or on integer factoring. Then in 2014, Mao et al. [62] gave the first lattice-based attribute signature whose hardness was based on the SIS problem, and it exploits the bonsai trees during signature generation. The details of the signature scheme are as below.

4.6.1. Signature Scheme

The signature generation consists of four algorithms $(A.KeyGen, A.Extract, A.Sign, A.Verify)$ which works as follows:
For integers m, q where $q = poly(n)$, $m \geq 6n \log q$ and the gaussian parameter $r = \tilde{L}\omega(\sqrt{n})$ where $\tilde{L} = O(\sqrt{n \log q})$, where n is the security parameter

- $A.KeyGen$: This algorithm is executed by a trusted party(TP) which on inputting the security parameter returns the master keys. TP chooses $2k$ random and independent matrices $\mathbf{A}_j^{(i)} \in \mathbb{Z}_q^{n \times m}$ where $j \in [k]$ and $i \in \{0, 1\}$. Then, it selects a collision resistant hash function $H : \{0,1\}^\star \longrightarrow \{0,1\}^t$, and then chooses $2t$ randomly and independent matrices $\mathbf{B}_j^{(i)} \in \mathbb{Z}_q^{n \times m}$ where $j \in [t]$ and $i \in \{0, 1\}$.

  Next, TP selects a vector randomly $\mathbf{u} \in \mathbb{Z}_q^n$. It then obtains $(\mathbf{A}, \mathbf{T}) \leftarrow TrapGen(1^n)$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$. At last $A.KeyGen$ returns the master public key $MPK = (\mathbf{A}, \mathbf{A}_j^{(i)}, \mathbf{B}_j^{(i)}, H, \mathbf{u})$ and the master secret key $MSK = (\mathbf{T})$

- $A.Extract$: This algorithm extract the private key of a signer with a set of specified attribute. For each attribute in the universe, if $w_j \in W$ then choose $\mathbf{A}_j^{(0)} \in \mathbb{Z}_q^{n \times m}$ else choose $\mathbf{A}_j^{(1)} \in \mathbb{Z}_q^{n \times m}$. Then set $\mathbf{A_u} = \mathbf{A}||\mathbf{A}_1^{(b)}|| \ldots \mathbf{A}_k^{(b)}$ where b depends on $w_j \in W$, now using $ExtBasis(\mathbf{A_u}, \mathbf{T})$ generate $\mathbf{T}'$. Randomize this basis to obtain a basis $\mathbf{T}_W$ which is the signing key for the user with W set of attribute.

- $A.Sign$: If a signer with a set of attribute W satisfying an access structure L wishes to sign the message M, then, he first computes $M' = H(M, L)$. Let m_i denote the ith bit of M'. The signer computes $\mathbf{A}_m = \mathbf{A_u}||\mathbf{B}_1^{(m_1)}|| \ldots ||\mathbf{B}_t^{(m_t)}$. Then, he generates the signature $\sigma \leftarrow SampPre(\mathbf{A}_m, ExtBasis(\mathbf{A}_m, \mathbf{T}_W), u, \delta)$ and outputs (M, L, σ).

- $A.Verify$: Verifier accept the signature iff $\|\sigma\| \leq r\sqrt{m(k + t + 1)}$ and $\sigma \in \mathcal{L}^\perp(\mathbf{A}_m)$.

4.6.2. Security Analysis

The above scheme is also proved to be secure against existential forgeability under the hardness of the SIS problem. Moreover, it is proved to have perfect privacy. But the scheme doesn't state what to do if there exist more than one person with the same attributes then how to find who signed or who can sign.

4.6.3. Remark

Similar to the above scheme, Lie et al. [55] introduced two different schemes in the standard model, and Zhang et al. in 2015 [93] also gave quantum resistance attribute-based signatures. But their schemes have drawbacks also being low in efficiency, results in long signatures and time-consuming. Another scheme introduced by Jia et al. [91] in 2016 developed attribute signature in the standard model, and they also worked on its efficiency. Table 6 compares the above schemes in terms of secret key size, public key size, and signature size. These papers don't encounter the question when there are more than one signers with the same attribute set; this can be a direction for future work along with some practical implementation of the proposed signature scheme. Moreover, all the above schemes utilize trapdoor, but one can try signature schemes without trapdoor, which can be more efficient and reduces the signature size.

Sig. Scheme	Secret key size	Public Key Size	Sign Size
[62]	$m^2 logq$	$(4k+2)mnlogq + nlogq + k$	$(2k+1)mlogq$
[55]	$m^2 logq$	$5mnlogq + k$	$3mlogq$
[93]	$m^2 logq$	$(k+2)mnlogq + k$	$3mlogq$
[91]	$m^2 logq$	$mnlogq + k + nklogq$	$mlogq + k$

Table 6: Comparison of Attribute Signature Scheme

5. Implementation

This section discuss the implementation results of basic lattice based signature schemes (with trapdoor and without trapdoor). Bansarkhani and Buchmann [10] gave the first software initiation of lattice signature [38] with the trapdoor introduced by [65]. The implementation was performed on Sun XFire 4400 server with 16 Quad-Core AMD opteron(tm) processor 8356 CPUs at 2.3 GHz, 64 GB and 64 bit Debian 6.0.6. The tables below states the implementation results of ring and matrix variant. Here in the table, ↑ implies that factor grows as n increases.

Running times [ms]										
		Keygen			Signing			Verification		
n	k	Ring	Mat	M/R	Ring	Mat	M/R	Ring	Mat	M/R
128	24	277	984	3.6	5	9	1.8	0.6	1.4	2.3
128	27	317	1,108	3.5	6	11	1.8	0.7	1.7	2.4
256	24	1,070	5,148	4.3	12	30	2.5	1.5	5	3.3
256	27	1,144	5,728	4.1	14	36	2.5	1.7	6	3.5
512	24	4,562	28,449	5.0	27	103	3.8	3	18	6
512	27	5,354	30,458	5.1	31	125	4.0	4	21	5.3
512	29	5,732	34,607	5.4	35	136	3.8	5	22	4.4
1024	27	28,074	172,570	6.0	74	478	6.4	10	97	9.7
1024	29	30,881	198,620	6.3	81	518	6.4	11	102	9.3
Improvement factor		30-190 ↑	10 -40 ↑	-	2-6 ↑	1.4 - 2 ↑	-	-	-	-

Sizes [kB]											
		Public Key			Secret Key			Pert. Matrix	Signature		
n	k	Ring	Mat	M/R	Ring	Mat	M/R	R and M	Ring	Mat	M/R
128	24	9.4	1200	128	4.4	528	163	257	5.8	5.3	0.9
128	27	11.8	1512	128	5.0	594	163	257	6.5	5.9	0.9
256	24	18.8	4800	256	9.8	2304	236	1026	12.5	11.4	0.9
256	27	23.6	6048	256	11.0	2592	236	1026	14.1	12.8	0.9
512	24	37.5	19,200	512	21.3	9984	469	4100	26.8	24.5	0.9
512	27	47.3	24,192	512	23.9	11232	470	4100	30.1	27.4	0.9
512	29	54.4	27,840	512	25.7	12064	470	4100	32.2	29.4	0.9
1024	27	94.5	96,768	1024	51.7	48384	936	16392	63.8	58.5	0.9
1024	29	108.8	111,360	1024	55.5	51968	936	16392	68.4	62.7	0.9
Improvement factor		-	-	-	-			170 - 260	-	-	-

Acti
Go to

Now, lattice based signature scheme without trapdoor [33] was also practically initiated on a desktop computer with intel core i7 at 3.4 GHz and 32 GB RAM with running openssl 1.0.1c. The table below states the results of implementation for 128, 160 and 198 bits.

Implementation	Security	Signature Size	SK Size	PK Size	Sign (ms)	Sign/s	Verify (ms)	Verify/s
BLISS-0	$\leqslant$ 60 bits	3.3 kb	1.5 kb	3.3 kb	0.241	4k	0.017	59k
BLISS-I	128 bits	5.6 kb	2 kb	7 kb	0.124	8k	0.030	33k
BLISS-II	128 bits	5 kb	2 kb	7 kb	0.480	2k	0.030	33k
BLISS-III	160 bits	6 kb	3 kb	7 kb	0.203	5k	0.031	32k
BLISS-IV	192 bits	6.5 kb	3 kb	7 kb	0.375	2.5k	0.032	31k

6. Conclusion

In this paper, we shed light on some different kinds of signature schemes that were developed over lattices. The article started with discussing the groundwork for lattice signature scheme and cited all the required assumptions that can be helpful for the rigorous study of basic concepts of lattices. Then it tries to cover the signature scheme proposed based on the hardness of lattice problems. Their working principle, security analysis, and the work carried out so far in the respective areas have been discussed, highlighting some of the future scopes.

7. Compliance with Ethical Standards

Author Sahadeo Padhye declares that he has no conflict of interest. Author Swati Rawal declares that she has no conflict of interest.
Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Abe M, Okamoto T. (2000) Provably secure partially blind signature. In proceedings of Crypto 2000, (pp. 271-286).

[2] Agrawal S , Boneh D, Boyen X (2010). Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In proceedings of CRYPTO 2010, (pp.98-115).

[3] Melchor C A, Cayrel P L, Gaborit P (2008). A new efficient threshold ring signature scheme based on coding theory. In proceedings of PQCrypto 2008, LNCS 5299, (pp.1-16).

[4] Aguilar M C, Bettaieb S, Boyen X, Fousse L, Gaborit P (2013). Adapting Lyubashevsky's signature schemes to the ring signature setting. In proceeding of AFRICACRYPT 2013, (pp. 1-25).

[5] Ajtai M (1996). Generating hard instances of lattice problems (extended abstract), In proceeding of 28th Annual ACM Symposium on the Theory of Comput. -STOC 96, (pp. 99-108).

[6] Ajtai M (1999). Generating hard instances of the short basis problem. In proceeding of ICALP 1999, (pp. 1-9).

[7] Alwen J and Peikert C (2009). Generating shorter bases for hard random lattices. In proceeding of STACS, (pp. 75-86).

[8] Babai L (1986). On Lovasz lattice reduction and the nearest lattice point problem. In proceeding of Combinatorica,6(1), (pp. 1-13).

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

[10] El Bansarkhani R., Buchmann J. (2014) Improvement and Efficient Implementation of a Lattice-Based Signature Scheme. In: Lange T., Lauter K., Lisoněk P. (eds) Selected Areas in Cryptography (SAC 2013). SAC 2013. Lecture Notes in Computer Science, vol 8282. Springer, Berlin, Heidelberg.

[11] Baum C, Damgard I, Oechsner S, Peikert C (2016). Efficient commitments and zero-knowledge protocols from Ring-SIS with applications to lattice-based threshold cryptosystems. IACR Cryptology ePrint Archive.

[12] Baum C, Lin H, Oechsner S (2018). Towards practical lattice-based one-time linkable ring signatures. In proceedings of Information and Communications Security. Springer International Publishing, Cham, (pp. 303-322).

[13] Bellare M, Micciancio D, Warinschi B (2003). Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In proceedings of EUROCRYPT 2003. (2656) (pp. 614-629).

[14] Bellare M, Rogaway P (1993). Random oracles are practical: A paradigm for designing efficient protocols. In proceedings of CCS, ACM.

[15] Bendlin R, Damgard I. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems (2010). In proceeding of TCC 2010. LNCS, 5978 (pp. 201-218).

[16] Bendlin R, Krehbiel S, Peikert C (2013). How to share a lattice trapdoor: threshold protocols for signatures and (H)IBE. In proceeding of ACNS 2013. LNCS, 7954 (pp. 218-236).

[17] Boneh D, Boyen X, Shacham H (2004). Short group signatures. In proceeding of CRYPTO 2004. LNCS, 3152 (pp. 41-55).

[18] Boyen X (2010). Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In proceeding of PKC 2010. LNCS 6056 (pp. 499-517).

[19] Brakerski Z, Kalai Y T (2010). A framework for efficient signatures,ring signatures and identity based encryption in the standard model. Cryptology Eprint Archive, Report 2010/086.

[20] Bresson E, Stern J, Szydlo M (2002). Threshold ring signatures and applications to ad-hoc groups.In proceedings of CRYPTO 2002. LNCS 2442, (pp. 465-480).

[21] Camenisch J, Neven G, Shelat A (2007). Simulatable adaptive oblivious transfer. In proceedings of EUROCRYPT 2007. LNCS 4515, (pp. 573-590).

[22] Camenisch J, Neven G, Ruckert M (2012). Fully Anonymous Attribute Tokens from Lattices. In proceeding of SCN 2012, 7485 (pp. 57-75).

[23] Cash D, Hofheinz D, Kiltz E (2009). How to delegate a lattice basis. In proceedings of CRYPTO 2009.

[24] Cash D, Hofheinz D, Kiltz E, Peikert C, Bonsai trees, or how to delegate a lattice basis. In proceedings of Eurocrypt 2010, LNCS 6110 (pp. 553-572).

[25] Cayrel P L, Lindner R, Ruckert M, Silva R (2010). Improved zero-knowledge identification with lattices. In proceedings of ProvSec 2010, LNCS 6402, (pp. 1-17).

[26] Cayrel P L, Lindner R, Ruckert M, Silva R (2010). A lattice-based threshold ring signature scheme. In proceeding of LATINCRYPT 2010. LNCS 6212 (pp. 255-272).

[27] Chaum D (1983). Blind signatures for untraceable payments. In proceeding of Advances in Cryptology 1983 (pp. 199-203).

[28] Chaum D, van Heyst, Eugene (1991). Group signatures. In proceeding of EUROCRYPT 1991, LNCS 547, (pp. 257-265).

[29] Wang C, QI M (2011). Lattice-based Proxy Signature Scheme. Journal of Information and Computational Science, , 8(12) (pp. 2451-2458).

[30] Cramer R, Damgard I, Schoenmakers B (1994). Proofs of partial knowledge and simplified design of witness hiding protocols. In proceeding of CRYPTO 1994. LNCS 839, (pp. 174-187).

[31] Desmedt Y (1987). Society and group oriented cryptography: a new concept. In Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology 1987, LNCS 293 (pp. 120-127).

[32] Whitfield D, Martin E H (1976). New directions in cryptography. IEEE Transactions on Information Theory, 1976, IT-22(6), (pp. 644-654).

[33] Ducas L, Durmus A, Lepoint T , Lyubashevsky V (2013). Lattice signatures and bimodal gaussians. In proceeding of CRYPTO 2013, LNCS 8042 (pp. 40-56).

[34] Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G , and Stehle D (2019). CRYSTALS-Dilithium, 2019). Available from: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/ Round-2-Submissions. Accessed on April 19, 2019.

[35] Fiat A, Shamir A (1986). How to prove yourself: Practical solutions to identification and signature problems. In proceedings of CRYPTO 1986. LNCS 263 (pp. 186-194).

[36] Gao W, Hu Y, Wang B, Xie J, Liu M (2017). Identity-based blind signature from lattices - Wuhan University Journal of Natural Sciences, 2017, 22(4) (pp. 355-360).

[37] Gordon S D, Katz J, Vaikuntanathan V (2010). A group signature scheme from lattice assumptions. In proceeding of ASIACRYPT 2010. LNCS 6477 (pp. 395-412).

[38] Gentry C, Peikert C, Vaikuntanathan V (2008). Trapdoors for hard lattices and new cryptographic constructions. In proceedings of STOC'08 Proceedings of the fortieth annual ACM symposium on Theory of computing 2008. (pp. 197-206).

[39] Wang F H, Hu Y P, Wang C X (2010). A lattice-based blind signature scheme. Geomatics and Information Science of Wuhan University, 2010, 35(5) (pp. 550-553).

[40] Jiang Y, Kong F, JU X (2010). Lattice-based Proxy Signature. In proceeding of Computational Intelligence and Security (CIS), (pp. 382-385).

[41] Kawachi A, Tanaka K, Xagawa K (2008). Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In proceedings of ASIACRYPT 2008, LNCS 5350 (pp. 372-389).

[42] Kim K S, Hong D, Jeong I R (2013). Identity-based proxy signature from lattices. Communications and Networks ,15(1)(pp. 1-17).

[43] Klein P N (2000). Finding the closest lattice vector when it's unusually close. In proceedings of SODA'00 Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms 2000. (pp. 937-941).

[44] Krawczyk H , Rabin T (2000). Chameleon signatures. In proceeding of Network and Distributed System Security Symposium NDSS, 2000.

[45] Laguillaumie F, Langlois A, Libert B, Stehle D (2013). Lattice-Based Group Signatures with Logarithmic Signature Size. In proceedings of ASIACRYPT 2013, LNCS 8270 (pp. 41-61).

[46] Langlois A, Ling S, Nguyen K, Wang H (2013). Lattice-Based Group Signature Scheme with Verifier-Local Revocation. In proceedings of PKC 2014, LNCS 8383, (pp. 345-361).

[47] YU L (2013). A Lattice-based Proxy Signature Scheme. Computer Engineering,39(0) (pp. 1-5).

[48] Libert B, Ling S, Nguyen K, Wang H (2016). Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors. In proceeding of EUROCRYPT 2016. LNCS 9666, (pp. 1-13).

[49] Libert B, Ling S, Mouhartem F, Nguyen K, Wang H (2016). Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In proceeding of ASIACRYPT 2016, LNCS 10032 (pp. 373-403).

[50] Libert B, Mouhartem F, Nguyen K (2016). A lattice-based group signature scheme with message-dependent opening. In proceeding of ACNS 2016, LNCS 9696 (pp. 137-155).

[51] Ling S, Nguyen K, Wang H. Group Signatures from Lattices: Simpler,Tighter, Shorter, Ring-Based. In proceeding of PKC 2015, LNCS 9020, (pp. 427-449).

[52] Ling S, Nguyen K, Wang H, Xu Y (2017). Lattice-based group signatures: Achieving full dynamicity with ease. In proceedings of ACNS 2017, LNCS 10355, (pp. 293-312).

[53] Ling S, Nguyen K, Wang H, Xu Y (2018). Constant-size group signatures from lattices. In proceedings of Public-Key Cryptography – PKC 2018. LNCS 10770 , (pp. 58-88).

[54] Ling S, Nguyen K, Wang H, Xu Y (2019). Lattice-based group signatures: Achieving full dynamicity (and deniability) with ease. In Theoretical Computer Science, 783, (pp. 71-94).

[55] Li M X, An N, Feng E Y (2015). An attribute-based signature scheme from lattices. Journal of Sichuan University (Engineering Science Edition), 47(2) (pp. 102-107).

[56] Lyubashevsky V (2008). Lattice-based identification schemes secure under active attacks. In proceedings of PKC 2008. LNCS 4939, (pp. 162-179).

[57] Lyubashevsky V (2009). Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In proceedings of Advances in Cryptology – ASIACRYPT 2009. LNCS 5912 (pp. 598-616).

[58] Lyubashevsky V (2012). Lattice signatures without trapdoors. In proceedings of, Advances in Cryptology – EUROCRYPT 2012, LNCS 7237 (pp. 738-755).

[59] Zhang L and Ma Y (2014). A Lattice-Based Identity-Based Proxy Blind Signature Scheme in the Standard Model Mathematical Problems in Engineering, Article ID 307637.

[60] Maji H K, Prabhakaran M, Rosulek M (2011). Attribute-based signatures. In Proceedings of the 11th International Conference on Topics in Cryptology (CT-RSA 2011), (pp. 376-392).

[61] Mambo M, Usuda K, Okamoto E (1996). Proxy signatures: delegation of the power to sign messages. IEICE Transactions Fundamentals 1996 (9) (pp. 1338-1353).

[62] Mao X P, Chen K F, Long Y (2014). Attribute-based signature on lattices. Journal of Shanghai Jiaotong University, 2014, 19(4) (pp. 406-411).

[63] Micciancio D and Goldwasser S (2002). Complexity of Lattice Problems: a cryptographic perspective. The Kluwer Inter-

national Series in Engineering and Computer Science.
[64] Micciancio D and Regev O. Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput., 37(1)(pp. 267-302).
[65] Micciancio D, Peikert C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller.In Proceedings of EUROCRYPT 2012. LNCS 7237 (pp. 700-718).
[66] Myers S, Sergi M, Shelat A (2011). Threshold fully homomorphic encryption and secure computation. IACR Cryptology ePrint Archive.
[67] Neumann J (1951). Various techniques used in connection with random digits. Research Nat. Bur.Stand., Appl. Math. Series, 1951(9) (pp. 36-38).
[68] Nguyen P Q, Zhang J, and Zhang Z (2015). Simpler Efficient Group Signatures from Lattices. In Proceedings of PKC 2015, LNCS 9020 (pp. 401-426).
[69] Peikert C (2009). Bonsai Trees: Arboriculture in Lattice-Based Cryptography. IACR Cryptology ePrint Archive.
[70] Peikert C (2010). An efficient and parallel Gaussian sampler for lattices. In Proceedings of CRYPTO, 2010, (pp. 80-97).
[71] Le H Q, Susilo W, Khuc T X, Bui M K, Duong D H (2019). A Blind Signature from Module Latices. In proceedings of IEEE Conference on Dependable and Secure Computing (DSC) 2019.
[72] Rawal S, Padhye S (2019). Threshold Ring Signature with Message Block Sharing. Security and Privacy. In proceedings of ISEA-ISAP 2019 CCIS(939) (pp. 1-9).
[73] Rawal S, Padhye S (2019). Cryptanalysis of ID based Proxy-Blind signature scheme over lattice. ICT Express , https://doi.org/10.1016/j.icte.2019.05.001 .
[74] Regev O (2005). On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of STOC, 2005 84-93.
[75] Regev O (2009). On lattices, learning with errors, random linear codes, and cryptography. ACM (JACM) 2009 56(6).
[76] Ruckert M (2010). Lattice-based blind signatures . In proceedings of 16th International Conference on the Theory and Application of Cryptology and Information Security , LNCS 6477,(pp. 413-430).
[77] Rivest R, Shamir A, Tauman Y (2001). How to leak a secret.In Proceedings of ASIACRYPT 2001, LNCS 2248, (pp. 552-565).
[78] Shor P (2006). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Comput. 26, (pp. 1484-1509).
[79] Shor P (1994). Algorithms for Quantum Computation: Discrete Logarithms and Factoring, In Proceedings of 35th Annual IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway,1994 (pp. 124-134).
[80] Tian M, Huang L. Breaking A Proxy Signature Scheme From Lattices. IJ Network Security, 14(6) (pp. 320-323).
[81] Tian H, Zhang F, Wei B (2016). A lattice-based partially blind signature - Security and Communication Networks, 9(12) (pp. 1820-1828).
[82] Torres A, W.A., et al (2018). Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1.0). In proceedings of ACISP 2018. LNCS(10946), (pp. 558-576).
[83] Wang F H, Hu Y P, Wang C X (2010). A lattice-based ring signature scheme from bonsai trees. J. Electron. Inf. Technol. 32(10), 2400.
[84] Wang J and Sun B (2011). Ring Signature Scheme from Lattice Basis Delegation. ICICS (pp. 15-28).
[85] Li W (2016). An Identity-Based Proxy Signature Scheme from Lattices in the Standard Model. In Proceedings International Conference on Intelligent Networking and Collaborative Systems 2016. DOI : 10.1109/INCoS.2016.17.
[86] Wang S, Zhao R, Zhang Y (2018). Lattice-based ring signature scheme under the random oracle model. International Journal of High Performance Computing and Networking 11(4), (pp. 332-341).
[87] Gao W, Hu Y, Wang B, et al., Improved lattice- based ring signature schemes from basis delegation. Journal of China Universities of Posts & Telecommunications, 23(3), (pp. 11-17) .
[88] Xia F, Yang B, Ma S (2011). Lattice-based Proxy Signature Scheme. Journal of Hunan University (Natural Sciences), 38(6) (pp. 84-88).
[89] Xie X, Xue R, Zhang R (2011). Efficient threshold encryption from lossy trapdoor functions. In Proceedings of PQCrypto 2011. LNCS 7071, (pp. 163-178).
[90] Lu X, Au M H, Zhang Z (2019). Raptor: A Practical Lattice-Based (Linkable) Ring Signature. In proceedings of Applied Cryptography and Network Security. ACNS 2019. LNCS (11464).
[91] Xie J, Hu Y, Gao J, Gao W, Li X (2016). Attribute-based signatures on lattices, The Journal of China Universities of Posts and Telecommunications, 23(4) (pp. 83-90).
[92] Yang C, Qiu P, Zheng S, Wang L (2015). An Efficient Lattice-Based Proxy Signature Scheme without Trapdoor.In Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2015.
[93] Zhang Y H, Hu Y P, Jiang M M (2015). An Attribute-based signature scheme from lattice assumption. Wuhan University Journal of Natural Sciences, 2015, 20(3) (pp. 207-213).
[94] Zhang L, and Sang Y (2012). A Lattice-based Identity-based Proxy Signature from Bonsai Trees. International Journal of Advancements in Computing Technology.
[95] Zhang L, Ma Y, and Sang Y (2013). A Lattice-based Multiple Grade Proxy Signature in the Standard Model. International Journal of Advancements in Computing Technology.
[96] Zhang L, Ma Y (2014). A Lattice-Based Identity-Based Proxy Blind Signature Scheme in the Standard Model. Mathematical Problems in Engineering.