GANITA,Vol.74(2), 2024, 65-77

A NEW R-LWE BASED DIGITAL SIGNATURE SCHEME

SWATI THAKUR ✉ **and HEMLAL SAHU**

Abstract

In cryptography, signature schemes are one of the most important cryptographic constructions, used to verify the sender's authenticity. In the presence of quantum computers, it has become obligatory to find more secure signature schemes that become good replacements for currently used signature schemes like RSA and ECDSA. Here we present a post-quantum lattice-based digital signature scheme that is based on the ring-learning With Errors(R-LWE) problem. To the best of my knowledge, this is the first signature scheme that works on the reconciliation system given by Peikert in 2014. In our signature scheme, we used uniform sampling during signature generation, such as GLP and ring-TESLA. We proved that our scheme is secure against chosen message attacks under the random oracle model. The security of our scheme is based on the SIS problem, and we have also analyzed the BKZ factor. We also claim that our signature scheme provides the required security in smaller key sizes than previous schemes.

2010 *Mathematics subject classification:* primary- 94A60; secondary-11P21.

Keywords and phrases: postquantum cryptography, signature, LWE.

1. Introduction

In the post-quantum era, lattice-based cryptography has become the most developing and interesting field of research in cyber security. Short Integer Solution (SIS) [2] and Learning With Errors (LWE) problem [24] are the two most important hard problems of lattice-based cryptography, they have proven security of worst-case to average-case reduction against quantum computers. They both provide a good platform for many cryptographic constructions based on them. Learning With Errors (LWE) problem provides strong security against quantum computers with versatility which increases its utility in cryptographic constructions. It is used in both classical as well as advanced constructions like a key exchange, encryption scheme, and signature scheme with other constructions like Fully Homomorphic Encryption (FHE), Attribute-based Encryption(ABE), Identity-Based Encryption (IBE), Oblivious Transfer, etc [6]. Here we are dealing with a signature scheme based on ring Learning with Errors (R-LWE). The main challenges with lattice-based signatures are larger key sizes of signature and public key and the requirement of sample from discrete Gaussian during the signing process, making it tough to implement.

The basic concept of the signature scheme is that the sender uses his private key to sign the message and send it to the receiver and the receiver verifies the sender's authentication by using his(sender's) public key. Initial approaches of lattice-based signature schemes were GGH and NTRU but both were broken by Nygun and Peikert [6]. First, a lattice-based signature scheme with proven security is the GPV-08 given by Gentry, Peikert, and Vaikuntanathan in which they used the trapdoor function to sign the message known as hash and sign methodology [14]. Another methodology mostly used in current signature schemes is the Fiat-Shamir paradigm which is secure in the random oracle model and was first used in the signature scheme by Lyubashevsky 2012 [18].

1.1. Related work In 2012, Lyubashevsky gave the first effective signature scheme based on SIS and learning With Errors problem without using the trapdoor function, in which he used the rejection sampling [18]. His signature scheme has many improvements and gave a good platform for many further signature schemes [1, 8, 12, 15]. Rejection sampling in lattice constructions was first used by Lyubashevsky in 2008 to construct a three-round identification scheme.

In the same year 2012, Lyubashevsky gave another signature scheme **GLP** with Guneysu and Poppelmann with some modification of his scheme which does not require Gaussian sampling and used compression technique for comparatively shorter signature size [15].

Bai and Galbraith gave another way of compression technique of Lyubashevsky's scheme and reduced the size of the signature. In the Lyubashevsky signing process two vectors y_1, y_2 were chosen with small norms and then computing $v = A.y_1 + y_2 \pmod{q}$ where A is $m \times n$ matrix and $B = A.S + E$ is a public key and S is the secret key. In Bai and Galbraith's Method they use only one vector y, instead of two vectors y_1, y_2 and they computed $v = A.y \pmod{q}$ and sign the message by using a hash function that reduced the signature size to 16500 bits to around 9000 to 12000 bits [8].

BLISS signature scheme was given by Ducas, Durmus, Lepoint, and Lyubashevsky in 2013, which is also based on the rejection sampling algorithm, which samples from a bimodal Gaussian distribution combined with a modified scheme instantiation. They have also claimed that their signature scheme has comparatively shorter signature and public key sizes than previously proposed lattice-based signature schemes [12].

In recent work Akleylek *et al.* have given another signature scheme named **TESLA-signature** [1], their scheme is an extension of the of Bai and Galbraith and its optimizations given by Dagdelen and their work was based on Ring-LWE, while Bai and Galbraith worked on matrix version of LWE. In their scheme two polynomials a_1, a_2 are chosen as public parameters and secret key s, and error functions e_1, e_2 are

sampled from Gaussian distribution, then t_1, t_2 are public keys where $t_i = a_i.s + e_i \pmod q$ for $i = 1, 2$ to sign message μ an independent vector is chosen uniformly polynomial $y \in R_q$ and compute $v_i = a_i.y \pmod q$ for $i = 1, 2$ and then signature $z = y + sc$ will be calculated, where $c = H(\lfloor v_1 \rceil_{d,q}, \lfloor v_2 \rceil_{d,q}, \mu)$ and then hash function is applied and for verification $c = H(\lfloor w_1 \rceil_{d,q}, \lfloor w_2 \rceil_{d,q}, \mu)$ where $w_i = a_i.z - t_i.c$ for $i = 1, 2$. They claim that their scheme is more efficient and with provable security and comparatively smaller key sizes [1].

1.2. Cryptographic construction *A* signature scheme contains four steps, setup, generation, signature, and verification process.

- Setup(pp)- If two parties want to use a digital signature scheme, then first they have to set some common parameters, known as public parameter pp.
- Gen(pk,sk)- They generate their own private key and public key and output the corresponding public key p_k and keep their private key secret.
- Sign (S_a,sk)- Sign the message μ with signing algorithm S_a with secret key s_k and out put the signature S^* with message μ.
- Verify (V_a, pk)- The receiver verifies the message using a verification algorithm with the sender's public key p_k.

$$V_a(\tilde{\mu}, S^*) = \begin{cases} true, & if\ S_a(\tilde{\mu}) = S^* \\ false, & \text{otherwise} \end{cases}$$

where $\tilde{\mu} = h(\mu)$ for $\mu \in M$

1.3. Organization In section 2, we will discuss briefly the notations and computational terminology required for our scheme. In section 3, we present our signature scheme. In section 4, we analyze the security measure and parameter theoretically. In section 5, we conclude our paper.

2. Preliminaries

In this section, we briefly examine the notation and basic background required for our cryptographic schemes based on ring learning With Errors(R-LWE) problem.

Let $q \in \mathbb{N}$ be a prime $\mathbb{Z}_q$ is integers modulo q and represented by set integers in range $(-q/2, q/2]$. we write vectors in boldface $\mathbf{v} = (v_1, \ldots, v_n)^T$ and Euclidean norm $\|v\|_2 = \sqrt{(\sum_{i=1}^{n} v_i^2)}$, $\|v\|_\infty = max|v_i|$ for $x \in R$ and $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor \in \mathbb{Z}$.

For an integer $q \geq 1$, let $\mathbb{Z}_q$ denote the quotient ring Z/qZ i.e., and $\mathbb{Z}_q[x]$ is polynomial ring modulo q and $\mathbb{R}_q = Z_q[x]/N$ is quotient ring where $N =< f(x) >$ is an ideal generated by irreducible polynomial, here taken the m^{th} cyclotomic polynomials of degree $n = 2^l$ which are irreducibles over rational, $R_q = \mathbb{Z}_q[X]/x^n + 1$

Elements of $\mathbb{R}_q$ are the polynomials of degree less than n that can be represented as $g(x) + < x^n + 1 >$ for some polynomial $g(x)$ whose degree can not exceed then $(n - 1)$. Multiplication of polynomials can be done by fast Fourier transform methods in $O(n. \log n)$ complexity.

2.1. Learning With Errors Problem(LWE) The learning with errors problem was introduced by O. Regev in 2005 [24]. It is parameterized by integers $n, q \in \mathbb{N}$ and distribution χ and ϕ on $\mathbb{Z}$, χ is uniform distribution on $\mathbb{Z}_q$ and $\phi = D_{\alpha,q}$ for some fixed real number $0 < \alpha < 1$.

Definition 2.1. let n, q be positive integers χ is a probability distribution on $\mathbb{Z}$ and $a \in \mathbb{Z}_q^n$ is choosing uniformly at random, choosing $e \in Z$ according to χ and $s \in \mathbb{Z}_q^n$ then $(a, b) = (a, a.s + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, if $A_{s,\chi}$ be the probability distribution on $Z_q^n \times \mathbb{Z}_q$ is called search-learning with error problem if recovering s from (a,c).

Decision-learning with errors(LWE) is the problem of deciding whether pair $(a, b) = (a, a.s + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, is sampled from $A_{s,\chi}$ or the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

2.2. Rejection sampling For security signature mustn't leak the secret key during signing, this insurance is given by rejection sampling introduced by Lyubashevsky [18]. Rejection sampling in lattice constructions was first used by Lyubashevsky in 2008 to construct a three-round identification scheme.

Let $f : Z^n \to R$ be a probability distribution, given a subset $U \leq Z^n$ and $h : U \to R$ be a probability distribution on V. Let $g_u : Z^n \to R$ be a family of probability distribution indexed by v such that for almost all v from h there exists a upper bound $M \in R$ such that

$$Pr[Mg_u(z) \geq f(z); z \leftarrow f] \geq 1 - \epsilon \tag{2.1}$$

Then the output distribution of the following has negligible statistical difference:

- $v \leftarrow h, z \leftarrow g_u$, output (z,v) with probability min $(\frac{f(z)}{Mg_u(z)}, 1)$.
- $v \leftarrow h, z \leftarrow f$, output (z,v) with probability $\frac{1}{M}$.

from the lemma [4.4, [18]] for $k > 0$, $Pr_{x \leftarrow D_\sigma}(|x| > k\sigma) \leq 2e^{-k^2/2}$, $k = 14$ gives the tail probability approximately 2^{-140}.

2.3. Discrete Gaussian If χ is a probability distribution over R, then $x \xleftarrow{s} \chi$ denotes sampling $x \in R$ according to χ. If S is a set then $U(S)$ denotes the uniform distribution on S and we denote sampling x uniformly at random from S either with $x \leftarrow U(S)$ $x \leftarrow S$ [21].

The distribution χ used in this paper is discrete Gaussian distribution (normal) on R. Discrete Gaussian to each $x \in \mathbb{Z}$ a probability proportional to $e^{-x^2/2(\sigma^2)}$ normalized with mean 0 and standard deviation as parameter σ, $\rho_\sigma(x) = e^{-x^2/(2\sigma^2)}$

$$\rho_\sigma(Z) = 1 + 2\sum_{k=1}^{\infty} e^{-k^2/2(\sigma^2)} \tag{2.2}$$

the discrete Gaussian distribution on $\mathbb{Z}$ is $D_{Z,\alpha}(x) = \rho_\sigma(x)/\rho_\sigma(Z)$ obtained by sampling each coefficient from $D_{Z,\alpha}(x)$ where [13]. $y \leftarrow D_\sigma^n$ denote that vector $y = (y_1, \ldots, y_n)$ in Z^n is independently sampled according to distribution D_σ.

2.3.1. ***Subgaussian*** For any random variable X over R is called δ-subgaussian with parameter $s > 0$ if for all $t \in R$ the moment generating function satisfies

$$E[exp(2\pi tX)] \leq exp(\delta).exp(\pi s^2 t^2) \tag{2.3}$$

for any $\delta \geq 0$ B bounded centered random variable X with property $E[X] = 0$ and $|X| \leq B$ is 0-subgaussian with parameter B $\sqrt{2}.\pi$, Thus simple bounded distributions, such as uniform random from an interval [-B, B], are 0-subgaussian with parameter $B\sqrt{2}\pi$.

Lemma 2.2. *If X_1 is a δ_1 subgaussian with parameter s_1 and X_2 is a δ_2 -subgaussian with parameter s_2 then $X_1 + X_2$ is a $\delta_1 + \delta_2$ subgaussian with parameter $\sqrt{s_1^2 + s_2^2}$ [22]*

.

2.4. Rounding and Reconciliation functions In 2014, Peikert introduced a new reconciliation system that reduced size by half. For an integer p that divides q, Peikert defines the modular rounding function and cross rounding function and its reconciliation mechanism [22].

Modular rounding function $\lfloor . \rceil_{q,2} : \mathbb{Z}_q \to \mathbb{Z}_2$

such that $\lfloor u \rceil_{q,2} = \lfloor \frac{2}{q}.u \rceil \bmod 2$

Cross rounding function $\langle . \rangle_2 : \mathbb{Z}_q \to \mathbb{Z}_2$

.

such that $\langle u \rangle_2 = \lfloor \frac{4}{q}.u \rfloor \bmod 2$

by the both rounding function the whole $\mathbb{Z}_q$ is divided in four subset I_0, I_1, I_0', I_1'

$$\begin{aligned} I_0 &= \{0, 1, \ldots, \lfloor \tfrac{q}{4} \rceil - 1\} \\ I_1 &= \{-\lfloor \tfrac{q}{4} \rfloor, \ldots - 1\} \\ I_0' &= (\tfrac{q}{2} + I_0) \\ I_1' &= (\tfrac{q}{2} + I_1) \end{aligned}$$

we see that rounding function $\lfloor u \rceil_2$ forms partitions of all elements $u \in Z_q$ [22]

$$\lfloor u \rceil_2 = \begin{cases} 0, & \text{if } I_0 \cup I_1 \\ 1, & \text{if } I_0' \cup I_1' \end{cases} \tag{2.4}$$

similarly, $\langle u \rangle_2$ also partition, all elements $u \in Z_q$ $\langle u \rangle_2 \in \{0, 1\}$

$$\langle u \rangle_2 = \begin{cases} 0, & \text{if } I_0 \cup (I_0') \\ 1, & \text{if } I_1 \cup (I_1') \end{cases} \tag{2.5}$$

Definition 2.3. **Reconciliation function** $\text{rec} : \mathbb{Z}_q \times \mathbb{Z}_2 \to \mathbb{Z}_2$

reconciliation function define as [22]

rec : $\mathbb{Z}_{2q} \times \mathbb{Z}_2 \to \mathbb{Z}_2$ with the set $E = [(-\frac{q}{8}, \frac{q}{8}) \cap Z]$

$$rec(w, b) = \begin{cases} 0, & \text{if } I_b + E \bmod 2q \\ 1, & \text{otherwise} \end{cases} \tag{2.6}$$

if $u, w \in \mathbb{Z}_q$ are sufficiently close, then $\lfloor u \rceil_2$ can be recovered from given w and $\langle u \rangle_2$ by using reconciliation function [22].

For odd q, and $u = w + e \in \mathbb{Z}_q$ where $u, w \in \mathbb{Z}_q$ and $e \in E$ then $rec(2w, \langle u \rangle_{2q,2}) = \lfloor u \rceil_{2q,2}$ [22]. Since we use odd q, in this paper we have to use the randomized doubling function to avoid biased, define as $dbl : Z_q \to \mathbb{Z}_{2q}$ as $dbl(x) = 2x - e$ where e is sampled from $\{-1, 0, 1\}$ with the probability of $p_0 = \frac{1}{2}$, $p_{-1} = p_1 = \frac{1}{4}$. It is 0 centered and therefore 0-subgaussian with parameter $\sqrt{2}\pi$.

both rounding and reconciliation functions can be extended to cyclotomic rings R using the decoding basis. If $D = \{x_j\}$ is decoding basis then $v = \sum_j v_j.x_j \in R_q$ for the coefficients $v_j \in r_q$ then $\lfloor u \rceil_2 = \sum_j \lfloor u_j \rceil_2.x_j$ similarly computation for $\langle u \rangle_2$ the reconciliation function rec: $R_q \times R_2 \to R_2$ can be obtained by $rec(w, b) = \sum_j (w_j, b_j).x_j$ where $w = w_j.x_j$, $b = b_j.x_j$ [22].

3. Our proposed signature scheme

In this section, we present our signature scheme, which is parameterized by integers $n, q, \sigma, \omega, B, U, \kappa$, where q is a prime number, such that $q = 1 \bmod 2n$ and we restrict here n to be the power of 2 and discrete Gaussian distribution D^n_σ is the probability distribution for sampling error and secret key with standard deviation σ and D^n_y and D^n_z are probability distribution for sampling y and z. D^n_y taken here uniform distribution over [-B, B] and D^n_z to be uniform distribution on [-(B-U), (B-U)]. It is beneficial to use uniform distribution over Gaussian distribution, which provides security against timing attacks [5].

The hash function $H : \{0, 1\}^* \to \{0, 1\}^\kappa$ to convert the string of any length to a fixed length and an encoding function F is encoding function close to be injective that maps binary strings of length κ to elements of $\mathbb{B}_{n,\omega}$ vector of length n and weight ω or $\mathbb{B}_{n.\omega} = \{v \in \{0, 1\}^n : \|v\|^2 = \omega\}$. We take on several ideas from [5, 8, 12] in parameter selection and security analysis of our signature scheme [5, 12].

Key generation:- Initially, a random polynomial a is chosen uniformly from R_q will be distributed to all considered as public parameters and choose two polynomials s, e_0 according to discrete Gaussian distribution such that the error is small enough (we set not more than ζ) to ensure that length of public key and signature are short where

$b = a.s + e_0 \in R_q$ is public key .

Algorithm 1: key generation
let $a \xleftarrow{s} R_q$ a is public parameter choose $s, e_0 \xleftarrow{s} D_\sigma$, $b = a.s + e_0 \in R_q$ b is public key and $s_k = s$ is secret key

TABLE 1. Proposed R-LWE Signature Scheme

Algorithm 2: Signing process	**Algorithm 3: Verification**
input(a, s, e_0, μ)	input((z, v_2, c)
step 1- 1. choose $y \leftarrow R_q$ 2. $v = a.y \in R_q$ 3. $\check{v} = dbl(v) \in R_{2q}$ 4. $v_2 = \langle \check{v} \rangle_{2q,2} \in \{0,1\}^n$ 5. $v_1 = \lfloor \check{v} \rceil_{2q,2}$ 6. $c = H\{v_1, \mu\}$ 7. $\tilde{c} = F(c)$ 8. $z = y + s\tilde{c}$ 9. output signature (z, v_2, c) with probability min $(\frac{D^n(z)}{M.D^n_{y,sc}(z)}, 1)$ 10. if $e_0.\tilde{c} \in E$ then confirm the signature otherwise go to step 1	11. $\tilde{c} = F(c)$ 12. $rec(2.(az - b\tilde{c}) \pmod q, v_2) = v'$ 13. $rec(2.w \pmod q, v_2) = v'$ 14. accept if $H(v', \mu) = c$ reject otherwise **Explanation**: $az - b\tilde{c} =$ $= a(y + s\tilde{c}) - (as + e_0)\tilde{c}$ $= ay + as\tilde{c} - as\tilde{c} - e_0\tilde{c}$ $= ay - e_0\tilde{c}$ $= v - e_0\tilde{c}$ $= w$

Signing Algorithm:- If "A" wants to create a signature for message $\mu \in \{0,1\}$, he chooses a polynomial y uniformly from R_q with uniform distribution over [-B, B] and then computes $v = a.y \in R_q$ and then he gets $\check{v}$ by applying doubling function to avoid the biased and apply modular rounding function and cross rounding function by which, v_1, and v_2 will be obtained. Since v_2 is used as a masking function of v_1 [22], will be sent to the receiver "B" and v_1 is used to create a signature by applying hash and encoding function ($c = H\{v_1, \mu\}$, $\tilde{c} = F(c)$) and "A " uses his private(signing) key to calculate $z = y + s\tilde{c}$ and will send signature (z, v_2, c) to "B". To check the correctness of the signature apply the rejection sampling compute $w = v - e_0.\tilde{c}$, if $|v - w| = e_0.\tilde{c} \leq q/8$ then signature (z, v_2) will sent to "B" otherwise restart from step 1.

Verification Algorithm:- To verify the signature after receiving (z, v_2, c), "B" first computes $\tilde{c} = F(c)$, assuming that both parties know about encoding function F before signature scheme as prior setup. Then he uses reconciliation function to recover v_1

from v_2 and $\tilde{c}$ by calculating $rec(2.(az - b\tilde{c}) \pmod q, v_2) = v'$, as $\lfloor \check{v} \rceil_{2q,2}$ can be recovered from $\langle \check{v} \rangle_{2q,2}$, i.e. $v_1 = v'$ up to correctness of 2^{-128} [7, 22], then the signature can be verified by using a hash function and a signature will be accepted if $H(v_1, \mu) = c$, otherwise signature will be rejected.

4. Security Analysis

In this subsection, we have proved that the signature scheme presented in this paper is unforgeable against a chosen-message attack in a random oracle model and secure as long as the learning with errors problem is computationally hard over rings.

THEOREM 4.1. *let $\mathcal{D}$ be a distinguisher that can query the random oracle H if he makes h queries to the random oracle and s queries to the signing algorithm then his advantage in distinguishing the actual signing algorithm and game 2 is at most $s(s+h).2^{-n+1}$.*

proof:- The only difference between the actual signing algorithm and the algorithm in game 1 is that in game 1, the output of the random oracle is chosen at random from $v : v \in \{0,1\}^{\kappa}$ programmed as $H(az - bc, \mu) = (ay, \mu)$ without checking whether the value was already in set when each time game 1 is called, the probability of generating a y such that ay is equal to one of the previous value as queried is at most 2^{-n+1} now at the most $s + h$. If we write public key as $A = [a, 1]$ then for any $t \in Z_{2q}^n$.

$$Pr[Ay = t; y \leftarrow D_{\sigma}^n] = Pr[y' = t - ay_0; y \leftarrow D_{\sigma}^n] \leq max \;\; Pr[y' = t'; y \leftarrow D_{\sigma}^n, t' \in Z_{2q}^n] \leq 2^{-n} \tag{4.1}$$

thus game 1 accessed s times and the probability of getting a collision each time is at most $(s + h).2^{-n+1}$ then the probability that a collision occurs in s queries is $s(s + h).2^{-n+1}$.

4.1. Random Oracle Model :- Let $\mathcal{F}$ be a forger against the signature scheme in a random oracle model that makes h random hash queries and $s$sign queries, runs in time t, and outputs a valid signature with probability δ.

Game 1 is the same as the original signature (Algorithm 1), except that sign queries are replaced by simulation in random oracle model

game 1:- (μ, a, s)
1. $y \leftarrow D_{\sigma}^{n}$ 2. choose uniformly binary string $c \leftarrow \mathbb{B}_{\omega}^{n}$ 3. $z \leftarrow sc + y$ 4. return (z,c) with probability min $(\frac{D^n(z)}{M.D_{y,sc}^{n}(z)}, 1)$ 5. program $H(az - bc, \mu) = c$
game 2:-(μ, a, s)
1. choose uniformly binary string $c \leftarrow \mathbb{B}_{\omega}^{n}$ 2. $z \leftarrow D_{\sigma}^{n}$ 3. return (z,c) with probability $\frac{1}{M}$ 4. program $H(az - bc, \mu) = c$

Case 1:- If there is a sign query on the message μ' with output (z', c) then $H(az - bc, \mu) = H(az' - bc, \mu')$ if $\mu \neq \mu'$ or $az - bc \neq az' - bc$ then there is a collision in H and this event occurs with probability $1/2^{\kappa}$ so it can be assumed that $\mu = \mu'$ and $az - bc = az' - bc$, therefore, $a(z - z') = 0$ if $z \neq z'$ then we have a non zero solution of the SIS instance or we have a solution of $az = 0$ for $\|(z - z')\| \leq 2(B - U)$.

Case 2:- Assume that c_j was a response to a random oracle query made by Forger, in this case, we record this signature (z, c_j) on the message μ and we generate random elements c_j, c_t and the By the General Forking lemma of Bellare and Neven we obtain that the probability $c'_j \neq c_j$ forger uses the random oracle in the forgery at least.

$$(\delta - \frac{1}{|\mathbb{B}_{\omega}^{n}|})(\frac{(\delta - \frac{1}{|\mathbb{B}_{\omega}^{n}|})}{t} - \frac{1}{|\mathbb{B}_{\omega}^{n}|}) \tag{4.2}$$

Thus with the above probability $\mathcal{F}$ output a forgery (z', c'_j) of the message μ and $(az - bc_j) = az' - bc'_j$

$$a(z - z') = b(c_j - c'_j) \tag{4.3}$$

but $z - z' \neq 0$ then

$$a(z - z') - (bc_j - bc'_j) = 0 \bmod 2q \tag{4.4}$$

$$a(z - z') - (as + e)c_j + (as + e)c'_j = 0 \bmod 2q \tag{4.5}$$

$$a(z - z' - sc_j + sc'_j) + e(c'_j - c_j) = 0 \bmod 2q \tag{4.6}$$

then we have a non-zero solution of SIS problem $ay_1 + y_2 = 0 \bmod q$ $y_1 = z - z' - sc_j + sc'_j$ and $y_2 = e(c'_j - c_j)$ if $(y_1, y_2) \neq (0, 0)$ then we have a solution of $ay_1 + y_2 = 0 \bmod q$ with $\|y\|_1 \leq 2B + 2\zeta\omega$ and $\|y\|_2 \leq 2\zeta\omega$.

Parameter selection

We aim to choose parameters to reduce the public key and signature key sizes while maintaining security and correctness. To do so, we follow the guidelines in section 4.4 in [22] and [18]. Security analysis done in [22] size of q, $q \geq n^3/2$. Following this we set $q = 25601$ for $n = 512$ and we choose ω such that $2^\kappa \geq 2^\omega \binom{n}{\omega}$ for security level 128 we take $\omega = 18$ and for security level 256 $\omega = 32$. And value of σ follow $\sigma \geq 2\sqrt{n}$ thus we choose 48, 52 value of σ for $n = 512$.

In order to apply the rejection sampling [18] lemma [4.4] follow $U = 14\sqrt{\omega}\sigma$ and $B = 14n\sqrt{\omega}\sigma$. So rejection probability will be $M = (\frac{2(B-U)+1}{2B+1})^n$

Table 2. parameters for signature scheme

security	n	q	ω	σ	B	U
128	512	25601	18	48	2^{21}	2852
256	512	40961	32	52	2^{22}	3802

BKZ-factor:-

To evaluate the security of our parameters against practical lattice attacks, the security can be estimated by root- Hermite factor γ of the lattices, for fixed $n, q, \sigma \geq 2\sqrt{n}$ we can compute the values (m, δ) satisfying the condition $\gamma = \gamma(m) = c\delta^m$ or $\gamma^{1/m} \geq \delta$ for $c = 1$ such that δ is maximal. As per standard security estimation, $\gamma \leq 1.0065$ should require 2^{128} operations to solve using BKZ lattice reduction.

We first compute $\beta = (q/\sigma).\sqrt{\ln(1/\varepsilon)/\pi}$ and then compute the root-Hermite factor $\delta = 2^{(\log^2 \beta)/(4n \log_2 q)}$, assuming that attacker can use optimal subdimension for this [19] $m = \sqrt{\frac{n \log(q)}{\log(\delta)}}$

running the BKZ lattice basis reduction algorithm given by

$$\log(t) = \frac{1.8}{\log \delta} - 110 \tag{4.7}$$

$n = 512$, $q = 25601$, for $\varepsilon = 2^{-128}$ security :-

for $\sigma = 8/\sqrt{2}.\pi$ we calculated the $\delta = 1.0049$

for $\sigma = 48 > 2\sqrt{n}$ we calculated the $\delta = 1.0026$

Our public size $n \log q$ bits and signature is depend on z and $v_2 = \langle v_1 \rangle_{2q,2}$ where size of z is $n \log 2(B-U)$ and v_2 is n therefore our signature size will be $n \log 2(B-U)+n+\kappa$. For $n = 512$ and 128-level security our scheme has 1424 byte signature size which is

the smallest size of previous schemes.

TABLE 3. Comparison of sizes

Name of scheme	public key	signature size	signature size for $n = 512$
BG	$2mn \log q$	$n\lceil \log 2(B-U)\rceil + \kappa$	11396 bits
R-Tesla	$2n\lceil \log q\rceil$	$n\lceil \log(2B-2U)\rceil + \kappa$	1488 byte
Our scheme	$n\lceil \log q\rceil$	$n\lceil \log 2(B-U)\rceil + n + \kappa$	1424 byte

5. Conclusion

We have proposed a new digital signature scheme based on ring learning with Errors problem. Which is based on Peikert's reconciliation system and proved secure against chosen message attacks under the random oracle model. Our signature scheme has smaller key sizes than previous schemes. Our signature scheme uses a uniform sampling signature which is safer against timing attacks rather than Gaussian sampling. The BKZ-Hermite factor has been analyzed by the parameter of our proposed scheme.

6. Acknowledgement

We are heartily thankful to Dr. B. K. Sharma for his valuable suggestion and we are also thankful to anonymous reviewers of the journal for their comments and corrections that improved this work.

References

[1] Akleylek S., Bindel N., Buchmann J., Krmer J., and Marson G. A.,*An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation*,Progress in Cryptology AFRICACRYPT 2016 LNCS **Springer**, vol 9646,(2016) 44-60.

[2] Ajtai M., *Generating Hard Instances of Lattice Problems*, Proceedings of the 28th Annual ACM Symposium on Theory of Computing STOC -96,(1996) 99-108.

[3] Albrecht M., Player R', and Scott S', *On the concrete hardness of Learning with Errors*, Journal of Mathematical Cryptology, 9(3)(2015) 169-203.

[4] Abdalla M., Fouque A. P., Lyubashevsky V. and Tibouchi M., *Tightly-Secure Signatures from Lossy Identification Schemes*, Journal of Cryptology, 29 (2013)597-631.

[5] Alkim E., Bindel N., Buchmann J., and Dagdelen O. *TESLA - Tightly-Secure Efficient Signatures from Standard Lattices*,Cryptology ePrint Archive, (2015) 755.

[6] Bernstein J., Buchmann J., and Dahmen E. *Introduction to post-quantum cryptography*,In:.(eds) Post-Quantum Cryptography Springer, Berlin, Heidelberg, (2009) 1-248.

[7] Bos W.J., Costello C., Naehrig M., and Stebila D., *Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem*,IEEE Symposium on Security and Privacy, (2015) 553-570.

[8] Bai S. and Galbraith D., *An Improved Compression Technique for Signatures Based on Learning with Errors*, Lecture Notes in Computer Science Springer, vol 8366,(2014) 28-47.

[9] Brakerski Z., Langlois A., Peikert C., Regev O., and Stehle D., *Classical Hardness of Learning With Errors*, Proceeding of 45 annual ACM symposium on the theory of computing, (2013)575-584.

[10] Bellare M. and Rogaway P., *The Exact Security of Digital Signatures-How to Sign with RSA and Rabin*, Advances in Cryptology EUROCRYPT-96 Lecture Notes in Computer Science, Springer, Berlin, Heidelberg vol 1070, (1996) 399-416.

[11] Dagdelen O., Bansarkhani R. E., Gpfert F., Gneysu T., Oder T., Pppelmann T and et. al., *High-speed signatures from standard lattices* Progress in Cryptology - LATINCRYPT 2014 Lecture Notes in Computer Science Springer, Heidelberg, vol 8895, (2014) 84- 103.

[12] Ducas L., Durmus A., Lepoint T. and Lyubashevsky V., *Lattice Signatures and bimodal Gaussians*, Advances in Cryptology CRYPTO 2013 Lecture Notes in Computer Science, vol 8042 (2013) 40-56.

[13] Dwarakanath N.C.and Galbraith D., *Sampling from discrete Gaussians for lattice-based cryptography on a constrained device*,Applicable Algebra in Engineering, Communication and Computing- AAECC 25, (2014) 159 -180.

[14] Gentry C., Peikert C., and Vaikuntanathan V., *Trapdoors for Hard lattices and New Cryptographic Constructions* Proceeding of the fortieth annual ACM symposium on the theory of computing,(2008)197- 206.

[15] Gneysu T., Lyubashevsky V., and Pppelmann T., *Practical Lattice-Based Cryptography A Signature Scheme for Embedded Systems* Cryptographic Hardware and Embedded Systems CHES 2012 Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol 7428, (2012)530-547.

[16] Lyubashevsky V., and Micciancio D., *Asymptotically Efficient Lattice-Based Digital Signatures*, *Theory of Cryptography TCC 2008 Lecture Notes in Computer Science* Springer, Berlin, Heidelberg, vol 4948,(2017)37-54, 2017.

[17] Lyubashevsky V., *Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures*, Advances in Cryptology ASIACRYPT 2009 Lecture Notes in Computer Science, Springer, Berlin, Heidelberg vol 5912, (2009) 598-616.

[18] Lyubashevsky V. *Signatures Without Trapdoors*, EUROCRYPT 2012 Lecture Notes in Computer Science Springer, vol 7237, pp 738-755, 2012.

[19] Lindner R., and Peikert C. , *Better Key Sizes (and Attacks) for LWE-Based Encryption*, Topics in Cryptology CT-RSA 2011 Lecture Notes in Computer Science Springer, Berlin, Heidelberg, vol 6558,(2010) 319-339.

[20] Lyubashevsky V., Peikert C., and Regev O., *On Ideal Lattices and Learning with Errors over Rings*, *Advances in Cryptology EUROCRYPT 2010 Lecture Notes in Computer Science* Springer, Berlin, Heidelberg vol 6110,(2010) 1-23.

[21] Lyubashevsky V., Peikert C., and Regev O., *A Toolkit for Ring-LWE Cryptography* Annual International Conference on the Theory and Applications of Cryptographic Techniques- EUROCRYPT 2013 Lecture Notes in Computer Science, Springer, Berlin, Heidelberg vol 7881, (2013)35-54.

[22] Peikert C., *Lattice Cryptography for the Internet*,PQCrypto 2014 Post-Quantum Cryptography, vol 8772, (2014) 197- 219.

[23] Peikert C., *A Decade of Lattice Cryptography*,Foundaion and Trends in Theoretical Computer Science 10(4),(2016) 283-424.

[24] Regev O., *On Lattices, learning with errors, random linear codes, and cryptography Journal of the ACM STOC* 56(6),(2009) 1-40.

[25] Regev O., *The Learning with Errors Problem*, (Invited Survey),IEEE 25th Annual Conference on Computational Complexity, (2010)191-204.

[26] Shor P., *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, 41(2),(1999) 303-332.

SWATI THAKUR, Govt. Naveen College, Nawagaon, sec-28, Naya Raipur, Raipur, India,
e-mail: thswati211@gmail.com

HEMLAL SAHU, Department of Mathematics, Govt. J. Y. Chhattisgarh College, Raipur, India
e-mail: hemlalsahu@gmail.com