Research Article
Efficient Lattice-Based Ring Signature Scheme without Trapdoors for Machine Learning

Qing Ye, Yongkang Lang, Zongqu Zhao, Qingqing Chen, and Yongli Tang

School of Software, Henan Polytechnic University, Jiaozuo 454000, China

Correspondence should be addressed to Zongqu Zhao; zhaozong_qu@hpu.edu.cn

Received 30 June 2022; Accepted 11 August 2022; Published 19 September 2022

Academic Editor: Le Sun

Copyright © 2022 Qing Ye et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine learning (ML) and privacy protection are inseparable. On the one hand, ML can be the target of privacy protection; on the other hand, it can also be used as an attack tool for privacy protection. Ring signature (RS) is an effective way for privacy protection in cryptography. In particular, lattice-based RS can still protect the privacy of users even in the presence of quantum computers. However, most current lattice-based RS schemes are based on a strong trapdoor like hash-and-sign, and in such constructions, there is a hidden algebraic structure, that is, added to lattice so that the trapdoor shape is not leaked, which greatly affects the computational efficiency of RS. In this study, utilizing Lyubashevsky collision-resistant hash function over lattice, we construct an RS scheme without trapdoors based on ideal lattice via Fiat–Shamir with aborts (FSwA) protocol. Regarding security, the proposed scheme satisfies unconditional anonymity against chosen setting attacks (UA-CSA), which is stronger than anonymity against full key exposure (anonymity-FKE), and moreover, our scheme satisfies unforgeability with respect to insider corruption (EU-IC). Regarding computational overhead, compared with other RS schemes that satisfy the same degree of security, our scheme has the highest computational efficiency, the signing and verification time costs of the proposed scheme are obviously better than those of other lattice-based RS schemes without trapdoors, which is more suitable for ML scenarios.

1. Introduction

Machine learning (ML) and privacy protection are inextricably linked. On the one hand, ML itself requires privacy protection, i.e., the training datasets and models in ML systems should not be disclosed. On the other hand, ML can also be an attack tool for privacy protection, and how to protect users’ sensitive information is a challenging task under the increasingly powerful ML technology.

Ring signature (RS), introduced by Rivest et al. [1] in 2001, is an effective technique for privacy protection in cryptography, which on the one hand enables the recipient of the data to believe that the source of the data is reliable, and on the other hand, obscures the identity of the data owner so that the recipient cannot be sure who is the real owner of the data. In an RS scheme, each user has a public key, and the signer can autonomously collect user’s public keys to form a ring without the permission or assistance of other users (where the signer is contained in the set of ring members), and the verifier only knows that the signature was generated under the ring, but is unsure which member’s private key is used as the signing key. Due to the anonymity of RS, it is widely used in fields such as e-voting, anonymous membership authentication, and anonymous tip-off.

Since the seminal work of [1], the research on RS has been unprecedentedly active. A general framework for constructing 1-out-of-n signature schemes was introduced by Abe et al. [2], which can use different types of keys to construct signatures based on integer decomposition and discrete logarithm problems. After that, many RS schemes along with their associated authentication schemes [3–10] have been proposed. However, the security definitions of these RS schemes are weak, i.e., they have not considered for certain realistic attacks. Bender et al. [11] proposed new definitions of anonymity and the unforgeability of RS to cover these attacks. Bender et al. [11] divided anonymity into three levels according to the degree of security, where the strongest version is anonymity against full key exposure.
(Anonymity-FKE), i.e., even if adversary is given the private keys of all ring members, the adversary is still unable to guess who is the genuine signer of the given RS. Regarding existential unforgeability, there are also three levels: unforgeability against fixed-ring attacks (EU-FRA), unforgeability against chosen-subring attacks (EU-CSA), and unforgeability with respect to insider corruption (EU-IC). Based on EU-CSA, the strongest EU-IC means the adversary cannot succeed in forging a signature, even if the adversary is allowed to obtain the private keys of ring members via asking a corruption oracle.

Most previous RS schemes [1–7, 9–11] are constructed under the assumptions of classical number theory, which are hardly resistant to quantum attacks [12]. In 1996, Ajtai [13] introduced the algebraic structure of lattice into cryptographic schemes. In the postquantum era, the new cryptosystem based on lattice has become a focus of research due to its merits of high asymptotic efficiency, simple operation, parallelizability, resistance to quantum attacks, and enjoying the average-to-worst reduction. The development of lattice-based provably secure encryption has developed rapidly and has made great progress [14–18], while lattice-based digital signature has experienced a tortured and bumpy process in the earlier years. First, Goldreich et al. [19] made an attempt at a lattice-based signature, then NTRU signature was proposed by Hoffstein et al. [20], and it was repaired and enhanced in [21, 22]. However, these digital signature schemes [19, 20, 22] on lattice failed to be proved secure under the attacks of [23, 24]. To date, provably secure signature schemes on lattice can be divided into two main branches: schemes with trapdoors and without trapdoors.

Digital signature schemes with trapdoors started from the “hash-and-sign” signature scheme constructed by Gentry et al. [25] based on SIS assumption, which is also the first lattice-based signature scheme, that is, provably secure. and Peikert [26] improved the structure of trapdoors, that is, proposed by [27], proposed the concept of G trapdoors on lattice, and remarkably improved the computational efficiency of trapdoor generation algorithm on lattice. A digital signature scheme on an ideal lattice was proposed by Ducas and Micciancio [28], and its trapdoor is constructed utilizing the technique of [26]. These trapdoors are deemed to be very suitable for lattice-based signature schemes, but the lattice is added with a hidden algebraic structure, which significantly affects the efficiency of signature schemes and is a payment that has to be considered. Digital signature schemes on lattice without trapdoors are mainly based on Stern’s zero-knowledge proofs. Although the Stern-type protocol is powerful, the soundness error of a single execution of the protocol is 2/3. Such protocol needs to be repeated many times so that the soundness error drops to a negligible value, so it is difficult to further improve their efficiency. To the best of our knowledge, only Lyubashevsky signature schemes [29, 30] without trapdoors are constructed not based on the Stern-type protocol, but on the Fiat–Shamir with aborts (FSwA) protocol, via which a very efficient digital signature scheme on lattice could be constructed.

Similar to a digital signature over lattice, lattice-based (linkable) RS can also be mainly divided into two branches, i.e., with trapdoors and without trapdoors. Lattice-based RS schemes with trapdoors have been extensively studied [31–36]. Notice that these lattice-based RS schemes with trapdoors, although are progressing about storage overhead, cannot always avoid the drawbacks brought by the structure of hash-and-sign, and the computational efficiency cannot be enhanced to a satisfactory level. Lattice-based RS schemes without trapdoors can be classified into membership proofs-based schemes and FSwA-based schemes. Several RS schemes [37–40] based on membership proofs have been proposed. Such RS schemes usually first construct a membership proof which is generally a zero-knowledge proof and then construct an RS scheme based on this proof. The efficiency of these RS schemes directly depends on that of underlying zero-knowledge proofs, which have the advantage that the signature length is usually O(log N), where N is the ring size, whereas the disadvantage is also obvious, i.e., the large and complex zero-knowledge proofs lead to low computational efficiency, and the length of RS could be large when N is small. (Linkable) RS schemes [41–44] based on lattice are efficient enough in terms of computational efficiency, with the drawback that the signature length is common O(N), yet it is very suitable in small-scale scenarios. For RS on lattice, several schemes [37–40] can satisfy the strongest Anonymity-FKE and EU-IC defined by Bender et al. [11]. In fact, Aguilar Melchor et al. [41] defined stronger anonymity, i.e., unconditional anonymity against chosen setting attacks (UA-CSA), and inspired by Lyubashevsky signature scheme [29], Aguilar Melchor et al. [41] constructed two RS schemes (AM1 and AM2) utilizing lattice-based collision-resistant hash function h ∈ H(D, Dx, m) [45], both of which can achieve UA-CSA for anonymity; regarding unforgeability, AM1 and AM2 satisfy EU-CSA and EU-IC, respectively. However, we deem that Aguilar Melchor et al. do not make good use of h ∈ H(D, Dx, m) in transforming Lyubashevsky digital signature [29] into RS, which causes the storage and computational overheads of both schemes of Aguilar Melchor et al. [41] to be large. In this work, via the FSwA protocol, we redesign an RS scheme on lattice without trapdoors using h ∈ H(D, Dx, m) again, the main contributions are as follows:

(1) Different from the RS schemes of Aguilar Melchor et al. [41] on lattice, in our key generation algorithm, the input value of h ∈ H(D, Dx, m) is taken as a user’s private key, the output value of h ∈ H(D, Dx, m) is taken as a public key, that is, relevant to the private key, which makes the length of a public key is reduced from a polynomial vector of m dimensions to a polynomial. Our signing algorithm is based on the framework of the RS scheme of Abe et al. [2], which is based on the discrete-log assumption, and the proposed scheme will be more concise and efficient.

(2) Under the improved security model of Aguilar Melchor et al. [41], the proposed RS scheme is rigorously proven to be safe. Regarding anonymity, our scheme satisfies the strongest UA-CSA; in terms of unforgeability, our scheme satisfies the strongest
EU-IC. And under the random oracle model, the unforgeability of the proposed scheme could be reduced to the approximate shortest vector problem (SVPx) over ideal lattice.

(3) Finally, with respect to performance and security, the proposed scheme is comprehensively compared with several schemes [37, 38, 40, 41]. The results show that compared with AM1 and AM2 of [41], the storage and computational overhead of our scheme are significantly reduced. In addition, compared with the computational overhead of other schemes, our scheme is remarkably superior, and is more suitable for ML applications.

2. Preliminaries

2.1. Notations. The symbol \([N]\) represents the abbreviation of the set \([1, 2, \ldots, N]\). Let \(\mathbb{R}\) be the set of real numbers, \(\mathbb{R}^+\) be the set of positive real numbers, and \(\mathbb{Z}\) be the set of integers. For a finite set \(S\), the symbol \(y \sim S\) denotes a random uniform sampling from \(S\). The upper-case letter \(X\) is the random variable denoting a signature, and \(X \leftarrow \mathcal{F}\) denotes \(X\) as the output of the signature algorithm \(\mathcal{F}\). Vectors and matrices are denoted by lower-case (e.g., \(x\)) and upper-case (e.g., \(X\)) letters in italic & bold, respectively. In this work, we construct a cryptographic scheme on ring \(D = \mathbb{Z}_q[x]/(x^n + 1)\), where \((x^n + 1)\) is an irreducible polynomial, and all logarithms we use are base 2. \(\mathbb{Z}_q\) is the set of integers modulo \(q\), elements in \(\mathbb{Z}_q\) are denoted by integers selected from the interval \([-q/2, q/2]\), and then the elements in \(D\) are represented by \(n - 1\) degree polynomials whose coefficients are taken from \(\mathbb{Z}_q\). For \(a = \sum_i^r a_i x^i \in D\), the common norms of \(a\) are given as follow:

\[
\begin{align*}
 l_1: \|a\|_1 &= \sum_i |a_i|, \\
 l_2: \|a\|_2 &= \left( \sum_i |a_i|^2 \right)^{1/2}, \\
 l_{\infty}: \|a\|_{\infty} &= \max_i |a_i|.
\end{align*}
\]

(1) For a polynomial vector \(a = (a_1, a_2, \ldots, a_m)\), where \(a_1, a_2, \ldots, a_m \in D, m\) is a positive integer, and the infinite norm with respect to \(a\) is defined as follows:

\[
\|a\|_{\infty} = \max_i \|a_i\|_{\infty}.
\]

Additionally, we will use the following notations:

\[
\begin{align*}
 D_c &= \{g \in D: \|g\|_{\infty} \leq 1\}, \\
 D_1 &= \{g \in D: \|g\|_{\infty} \leq \sqrt{n} \log n\}, \\
 D_x &= \{g \in D: \|g\|_{\infty} \leq mn^{1.5} \log n + n \log^2 n\}, \\
 D_y &= \{g \in D: \|g\|_{\infty} \leq mn^{1.5} \log n\}, \\
 D_z &= \{g \in D: \|g\|_{\infty} \leq mn^{1.5} \log n - n \log^2 n\},
\end{align*}
\]

(3) The complexity of algorithms is measured using the standard asymptotic notations \(\omega, O\):

\[
\begin{align*}
f(n) &= \omega(g(n)) \Rightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty, \\
f(n) &= O(g(n)) \Rightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} \neq \infty.
\end{align*}
\]

(4) Use the symbol \(\tilde{O}\) to suppress poly-logarithmic factors, and for example, for any constant \(c\) and \(c'\), \(f(n) = O(n^{c'} \log^c n)\), we have \(f(n) = O(n^{c'})\).

2.2. Lattice and Ideal Lattice. Micciancio [46] first proposed the concept of cyclic lattice, which to a certain extent eliminated the drawbacks of big key size and operational inefficiency of cryptographic schemes on Euclidean lattice. Lyubashovsky and Micciancio [45] first proposed the definition of an ideal lattice, which is a lattice with a special algebraic structure and is a generalization of the cyclic lattice. In general, a Euclidean lattice is a subgroup of a group, and an ideal lattice is an ideal of a ring.

Definition 1 (lattice). Supposing matrix \(B\) is composed of a set of linearly independent vectors \(b_1, b_2, \ldots, b_m \in \mathbb{Z}^n\), then the integer lattice generated by \(B\) is defined as:

\[
\mathcal{L}(B) = \{Bx : x \in \mathbb{Z}^m\}.
\]

(5) \(b_1, b_2, \ldots, b_m\) form a set of bases over \(\mathcal{L}(B)\). In short, \(\mathcal{L}(B)\) is a discrete additive subgroup in the \(n\)-dimensional real space \(\mathbb{R}^n\).

Definition 2 (\(q\)-ary lattice). Given a prime \(q\), positive integers \(m, n\), for any matrix \(A \in \mathbb{Z}_q^{m \times n}\), the integer lattice, that is, \(m\)-dimensional full-rank is described as follows:

\[
\Lambda^+(A) = \{u \in \mathbb{Z}^m \text{ s.t. } Au = 0 \pmod{q}\}.
\]

(6) For \(f\) to be a monic irreducible polynomial of degree \(n\), the ideal \(I \subseteq \mathbb{Z}[x]/(f)\), then an integer lattice \(\mathcal{L}(B) \subseteq \mathbb{Z}^n\) such that \(\mathcal{L}(B) = \{g \mod f : g \in I\}\) is called an ideal lattice.

2.3. Hardness Assumption

Definition 4 (R-SIS_{q,m,β} problem). Given a randomly chosen vector \(a \in D^m\), find the vector \(x \in D^m\) such that \(a^T \cdot x = 0\) satisfying \(0 < \|x\|_{\infty} \leq \beta\), where \(q, m\) are positive integers, \(\beta \in \mathbb{R}^+\).

Definition 5 (SVPx). Given a lattice \(\mathcal{L}(B)\) and an approximation factor \(\gamma \geq 1\), find a nonzero vector \(v\) on lattice such that \(\|v\|_{\infty} \leq \gamma\|u\|_{\infty}\) holds for any vector \(u \in \mathcal{L}(B)\).

2.4. Collision-Resistant Hash Functions and Bounding \(\|ac\mod(x^n + 1)\|_{\infty}\). Lyubashovsky and Micciancio [45] defined a family of collision-resistant hash functions on ideal lattice, which are efficient functions based on the hardness of worst-case lattice problems, and showed that finding
collisions for \( h \in \mathcal{H}(D, D_x, m) \) is at least as hard as solving SVP, where \( \gamma = \tilde{O}(n^2) \).

**Definition 6** (family of hash functions). For any integer \( m \) and set \( D_x \), the family of hash functions \( \mathcal{H}(D, D_x, m) = \{ h_a: a \in D^m \} \), where a hash function \( h_a: D^m \rightarrow D \). On input polynomial vectors \( b = (b_1, b_2, \ldots, b_m) \in D^m \), then \( h_a(b) = a \cdot b = a_1 b_1 + a_2 b_2 + \cdots + a_m b_m \). There are two homomorphic properties about \( h \in \mathcal{H}(D, D_x, m) \) as follows:

\[
\begin{align*}
    h(x + y) &= h(x) + h(y), \\
    h(xc) &= h(x)c,
\end{align*}
\]

where \( x, y \in D^m, c \in D \).

**Definition 7** (collision problem \( \text{Col}(h, D_x) \)). Given a function \( h \in \mathcal{H}(D, D_x, m) \), find two distinct vectors \( z_0, z_1 \in D_x \) such that \( h(z_0) = h(z_1) \).

The following lemma shows that when \( D_x \) is some limited domain (e.g., a set of small norm polynomials), solving the \( \text{Col}(h, D_x) \) problem is as hard as solving SVP, in the worst case on the lattice corresponding to the ideal \( \mathfrak{I}D_x \).

**Lemma 1** (see [29]). Let \( n \) be any power of 2, \( \|g\|_\infty \leq d \), and define the set \( D_x = \{ g \in D: \|g\|_\infty \leq d \} \), and \( q \geq 4dn^{1.5} \log n \). Then, for any \( h \in \mathcal{H}(D, D_x, m) \), if there exists a polynomial-time algorithm that solves \( \text{Col}(h, D_x) \) with some non-negligible probability, then there exists a polynomial-time algorithm that can solve SVP, over ideal lattice, where \( \gamma = 16dn^2 \log^2 n \).

In addition, we recall the following boundary Lemma to justify the security of the proposed scheme.

**Lemma 2** (Lemma 2.11 in [47] applied to our setting). Let \( \mathcal{L} \) be the response returned by the random oracle \( H \) (i.e., \( c \leftarrow D_x \)), then

\[
\Pr \left[ \|a\|_\infty \leq n\log^2 n \right] \geq 1 - 4e^{-\log^2 n/8} = 1 - n^{-\omega(1)}.
\]  

### 2.5. Statistical Distance and Probabilistic Lemma

The difference between two probability distributions can be measured by the statistical distance, and in the security proof of the proposed RS scheme, we complete the proof of anonymity by using it.

**Definition 8** (statistical distance). For random variables \( X_0 \) and \( X_1 \) that are defined on a countable set \( S \), if the set \( S \) is discrete, then the statistical distance between \( X_0 \) and \( X_1 \) is described as

\[
\Delta(X_0, X_1) = \frac{1}{2} \sum_{x \in S} \Pr \left[ X_0 = x \right] - \Pr \left[ X_1 = x \right].
\]

If the random variables \( X_0 \) and \( X_1 \) satisfy \( \Delta(X_0, X_1) \leq n^{-\omega(1)} \), then \( X_0 \) and \( X_1 \) are statistically close.

Regarding statistical distances, there are the following common properties:

1. \( \Delta(X_0, X_1) \geq 0; \)
2. \( \Delta(X_0, X_1) = \Delta(X_1, X_0). \)

In addition, to prove the convergence of our algorithms, the following probabilistic lemma will be used:

**Lemma 3** (Corollary 6.2 in [47]). For any \( s \in D^m \),

\[
\Pr_{z \leftarrow D, x \leftarrow D^m} \left[ |x + s| \leq 1 - o(1) \right] = 1.
\]

### 3. Definition of RS and Security Model

#### 3.1. Definition of RS

Suppose there exist \( N \) members in the ring and each member \( U_i (i \in [N]) \) has a pair of keys, i.e., public key \( pk \) and private key \( sk \), a RS scheme can be composed of four polynomial-time algorithms:

1. **Setup(1\(^\lambda\))**: Taking a security parameter \( \lambda \) as input, it outputs the public parameters \( pp \).
2. **KeyGen(pp)**: Taking the public parameters \( pp \) as input, it generates a pair of keys \( (pk, sk) \), where \( pk \) and \( sk \) denote the public and private keys, respectively.
3. **Sign(pp, Lpk, \mu, sk)\_\mu**: On input public parameters \( pp \), ring \( L_{pk} \), message \( \mu \) and the private key \( sk \) of the signer \( U_{\mu} \) (require that its corresponding public key \( pk_{\mu} \in L_{pk} \)), it outputs an RS sig of \( U_{\mu} \) on the message \( \mu \) under ring \( L_{pk} \).
4. **Verify(pp, Lpk, \mu, sig)**: On input public parameter \( pp \), ring \( L_{pk} \), message \( \mu \) and RS sig. If sig satisfies the verification conditions, it outputs 1; otherwise, it outputs 0.

#### 3.2. Security Model

For the above algorithms, an RS scheme is called to be secure if it satisfies the following definitions: correctness, anonymity, and unforgeability.

**Definition 9** (correctness). If the signer signs honestly, i.e., according to the algorithms in Section 3.1, the Verify algorithm will always output 1 with overwhelming probability, that is, equality (11) holds.

\[
\Pr \left[ \text{Verify}(pp, L_{pk}, \mu, sig) = 1 \right] = 1 - \negl(\lambda).
\]

Bender et al. [11] proposed the security definitions of RS under different security degrees, among which the highest degree in terms of anonymity is anonymity-FKE. Based on anonymity-FKE, Aguilar Melchor et al. [41] proposed a stronger definition, namely UA-CSA. In this work, the
security proof of anonymity is based on the security model of UA-CSA.

Let’s define a game between adversary \( \mathcal{A} \) and challenger \( S \). Under UA-CSA since all secrets are known, the adversary can effectively simulate the signature and corrupt oracles, so these two oracles are no longer provided in the game and the procedure in which \( S \) generates \( pp \) and \( (pk, sk) \) is not required, since they will be generated by the adversary instead. The game is as follows:

(i) Suppose \( l \) is the upper bound of ring member size in the system, the adversary submits to a set of public parameters \( pp \), ring \( L_{pk} = \{pk_1, pk_2, \ldots, pk_N\} \), two private keys \( sk_{i_1}, sk_{i_2} \) message \( \mu \), where \( N, i_1, i_2 \in [l] \).

(ii) \( S \) randomly selects \( b \sim \{0, 1\} \), invokes \( \text{Sign}(pp, L_{pk}, sk_b) \) to generate the signature \( \text{sig}_{0_b} \), and returns \( \text{sig}_{0_b} \) to \( \mathcal{A} \).

(iii) Adversary \( \mathcal{A} \) outputs a bit \( b' \), and if \( b' = b \), then \( \mathcal{A} \) wins the game.

The advantage of winning the game is denoted by,

\[
Adv_{\mathcal{A}}^{\text{anon}} = \Pr[b' = b] - \frac{1}{2}.
\]  

**Definition 10** (anonymity). An RS scheme satisfies UA-CSA if \( Adv_{\mathcal{A}}^{\text{anon}} \) is negligible for any polynomial-time adversary \( \mathcal{A} \).

Bender et al. [11] consider two cases when defining the strongest unforgeability EU-IC: (1) An adversary can trick some honest user into using the public keys that are adaptively generated to generate a signature. (2) The adversary can adaptively corrupt some ring members and obtain their keys. The strongest EU-IC with respect to RS is depicted by the game between adversary \( \mathcal{A} \) and challenger \( \mathcal{A}' \) as follows:

(i) Setup phase: Given security parameter \( \lambda \), \( \mathcal{A}' \) runs the KeyGen algorithm to generate the user’s public/private key pairs, and sends public parameters \( pp \) and the maximum set of user’s public keys \( S = (pk_i)_{i \in [l]} \) to adversary \( \mathcal{A} \).

(ii) Query phase: \( \mathcal{A} \) is allowed to make adaptive queries to the signature oracle \( \text{SO} \) and the corrupt oracle \( \text{CO} \).

(a) Signing query: \( \mathcal{A} \) submits to \( \mathcal{A}' \) the set of user’s public keys \( L_{pk} \), message \( \mu \) and \( \pi \in [l] \), where \( \pi \) is an index such that \( pk_{\pi} \in L_{pk} \). On receiving \((\pi, \mu, L_{pk})\), \( \mathcal{A}' \) invokes the Sign algorithm to generate a signature and send it to \( \mathcal{A} \).

(b) Corruption query: \( \mathcal{A} \) submits \( i (i \in [l]) \) to access corrupt oracle \( \text{CO} \), then \( \mathcal{A}' \) returns the relevant private key \( sk_i \) to \( \mathcal{A} \).

(iii) Forgery phase: \( \mathcal{A} \) outputs \( (\mu^*, L_{pk}^*, \text{sig}^*) \), and we call \( \mathcal{A} \) gets the triumph if the below conditions are satisfied:

(a) Verify \((pp, L_{pk}^*, \mu^*, \text{sig}^*) = 1; \)

(b) \( \mathcal{A} \) never queried for the signature on \((*, \mu', L_{pk}^') \);

(c) \( L_{pk}^* \subset S \) \( C \), where \( C \) is the set of corrupted users.

The advantage of \( \mathcal{A} \) winning the game is depicted as:

\[
Adv_{\mathcal{A}}^{\text{forge}} = \Pr[\mathcal{A} \text{ wins the game}].
\]

**Definition 11** (unforgeability). A RS scheme is said to satisfy EU-IC if \( Adv_{\mathcal{A}}^{\text{forge}} \) is negligible for any polynomial-time adversary \( \mathcal{A} \).

### 4. Lattice-based RS Scheme without Trapdoors

#### 4.1. Construction

(i) Setup \((1^\lambda)\): Given security parameter \( \lambda \), it outputs the public parameters \( pp = \{n, d, q, m, H, h, D_y, D_x, D_y, D_z\} \), where integer \( n > \lambda \) and is a power of \( 2, d = \sqrt[4]{\log n} \), prime \( q \geq 4(m^{\log \log n})^2 \), integer \( m > \log q / \log 2 \), \( H : \{0, 1\}^* \rightarrow D_y \) is a cryptographic hash function, \( h \) is randomly chosen from the family \( \mathcal{H}(D_y, D_x, m) \), and the other parameters are defined in Section 2.1.

(ii) KeyGen \((pp)\): Given public parameters \( pp \), it picks \( r \sim D_y^\mu \), then generates the public key \( pk = h(r) \) and sets the private key \( sk = r \).

(iii) Sign \((pp, L_{pk}, \mu, sk)\): Given public parameters \( pp \), ring \( L_{pk} = \{pk_1, pk_2, \ldots, pk_N\} \), message \( \mu \in [0, 1]^* \) and private key \( sk = r \) of member \( U_i \), it runs as follows:

(a) Choose \( r_{z,x} \sim D_y^\mu \),

(b) Calculate \( t_1 = h(r_{z,x}) - c_i pk_i, \ c_i = (\mod N + 1) \)

(c) Calculate \( r_{z,x} = u + c_i r_{z,x} \);

(d) If \( r_{z,x} \notin D_y^\mu \), then return to step (i);

(e) Output the signature \( \text{sig} = (c_1, (r_{z,x})_{i \in [N]}) \).

(iv) Verify \((pp, L_{pk}, \mu, \text{sig})\): Given public parameters \( pp \), ring \( L_{pk} = \{pk_1, pk_2, \ldots, pk_N\} \), message \( \mu \in [0, 1]^* \) and signature \( \text{sig} = (c_1, (r_{z,x})_{i \in [N]}) \), it accepts the signature and outputs 1 only if sig satisfies the below conditions, otherwise it rejects the signature and outputs 0.

(a) For \( i \in [N] \), \( r_{z,x} \notin D_y^\mu \);

(b) For \( i \in [N] \), calculate \( t_1 = h(r_{z,x}) - c_i pk_i, \ c_i = (\mod N + 1) \)

(c) Determine \( c_1 = H(L_{pk}, \mu, t) \), and determine \( c_i = H(L_{pk}, \mu, t) \).

#### 4.2. Correctness and Convergence of the Scheme

**Theorem 1** (correctness). The proposed RS scheme satisfies correctness.
Proof. When sig is a signature generated according to the signature algorithm, then (a) in step (ii) and step (iv) of Sign ensure that $(r_{z_i})_{i \in \{N\}}$ are elements in $D_z^n$. In addition, we have the following equations:

$$c_2 = H(L_{pk}, \mu, t_1), \text{ where } t_1 = h(r_{z_1}) - c_1 pk_1,$$

$$c_3 = H(L_{pk}, \mu, t_2), \text{ where } t_2 = h(r_{z_2}) - c_2 pk_2,$$

$$\vdots$$

$$c_n = H(L_{pk}, \mu, t_{n-1}),$$

where $t_{n-1} = h(r_{z_{n-1}}) - c_{n-1} pk_{n-1},$

since $r_{z_n} = u + c_n x_n \in D_z^n$, we have

$$c_{n+1} = H(L_{pk}, \mu, t_{n+1}) = H(L_{pk}, \mu, h(r_{z_n}) - c_n pk_n)$$

$$= H(L_{pk}, \mu, h(u + c_n r_n) - c_n pk_n)$$

$$= H(L_{pk}, \mu, h(u)) + c_{n+2}$$

$$= H(L_{pk}, \mu, t_{n+1}), \text{ where } t_{n+1}$$

$$= h(r_{z_{n+1}}) - c_{n+1} pk_{n+1} + c_{n+2}$$

$$= H(L_{pk}, \mu, t_{n+2}),$$

where $t_{n+2} = h(r_{z_{n+2}}) - c_{n+2} pk_{n+2},$

$$\vdots$$

$$c_1 = H(L_{pk}, \mu, t_N), \text{ where } t_N$$

$$= h(r_{z_N}) - c_N pk_N.$$  

(14)

To sum up, the polynomial sequence $\{c_1, c_2, \ldots, c_N\}$ in the verification process is equal to that in the signature process, so it must pass the verification of the Verify algorithm, and the proposed scheme is correct. □

Theorem 2 (convergence). Under the parameter settings of algorithm Setup, the expected runtime of the proposed scheme is $O(n)$, and algorithm Sign is expected to repeat no more than three times.

Proof. The proposed RS scheme is made up of four algorithms: Setup, KeyGen, Sign and Verify. The Setup algorithm selects a hash function $h \in H(D, D_x, m)$ (i.e., pick $mn$ random numbers from $\{-(q-1)/2, -(q-1)/2 + 1, \ldots, (q-1)/2\}$), the time for the step is negligible. The step of generating a single user's private/public keys in KeyGen is to randomly select a vector $r \sim D_z^n$, which simply involves randomly selecting $mn$ numbers from the set $\{-\sqrt{n \log n}, \ldots, \sqrt{n \log n}\}$, and then calculating $pk = h(r)$, which takes $O(n)$ time according to Lemma 2.16 from [47]. The Sign algorithm is to randomly select a vector $u \sim D_z^n$ and $N-1$ vectors $r_{z_i} \sim D_z^n$, then calculate small polynomial multiplication $c_i pk_i$ and hash function $h \in H(D, D_x, m)n$ times, and access random oracles $N$ times. The time of Sign algorithm running once is $O(n)$ from [47], but if $r_{z_n} \notin D_z^n$, then the operations of Sign need to be repeated again. Lemma 3 states that for any $r \sim D_z^n$, $Pr_{u \sim D_z^n, v \sim D_z^n}[u + vr \sim D_z^n] = 1/\alpha(t)$. Therefore, we will iterate Sign no more than three times and the runtime of Sign is also $O(n)$. Finally, the Verify algorithm needs to calculate small polynomial multiplication $c_i pk_i$ and function $h \in H(D, D_x, m)n$ times, and access random oracles $N$ times, thus the running time is also $O(n)$. □

5. Security

5.1. Anonymity. Before proving the anonymity of the proposed scheme, we first give and prove the following lemma, which shows that for an adversary who has the ability to distinguish two ring signatures based on adversarially-chosen private keys and the corresponding $c_i$ associated with the private keys, the statistical distance between the following two sets of random variables $Y_0$ and $Y_1$ is negligible.

Define two sets of random variables $Y_0 = (a_i \in D_z^n ; i \in [N], b)$ and $Y_1 = (a_i \in D_z^n ; i \in [N], b)$ that are obtained from the Sign algorithm with input $(pp, L_{pk}, \mu, sk_i)$ and $(pp, L_{pk}, \mu, sk_i)$, where the first $N$ components of $Y_b$ represent the first $N$ outputs $(r_{z_i})_{i \in [N]}$ of sig$_{b_0}$, and the $(N+1)$-th component of $Y_b$ represents $c_i$ corresponding to sig$_{b_0}$, $b \sim [0, 1]$. In addition, we use $Y_{i}^{(j)}$ to represent the $i$-th component of $Y_{b_0}$, $i \in [N + 1]$, e.g., $Y_{0}^{(N+1)}$ represents $c_x$.

Lemma 4. If $Y_0$ and $Y_1$ are random variables obtained from two legitimate signatures, and these two legitimate signatures are generated by private keys sk$_{b_0}$, sk$_{b_1}$ which are adversarially-chosen, we have

$$\Delta(Y_0, Y_1) = n^{-w(1)}.$$  

(15)

Proof. First define a set $D_i = \{d \in D_i : \|sk_i d\|_{\infty} \leq n \log^2 n\}$. According to Lemma 2, it is concluded that almost all elements of $D_i$ are in $D_i = \{sk_i d \in \mathbb{Z} \}$. Even if the private keys sk$_{b_0}$, sk$_{b_1}$ are chosen by the adversary, Lemma 2 will also guarantee that $|D_i {. sk_i} | \leq |D_i| = 1 - n^{-w(1)}$, where $n^{-w(1)}$ is a negligible function. Then divide the statistical distance $\Delta(Y_0, Y_1)$ into two parts, we have formulas (16) and (17).

Next we will discuss $\Delta(Y_0, Y_1)$ in two steps, first to prove that formula (16) is negligible (Step 1), and then to prove that formula (17) is equal to zero (Step 2).
\[
\Delta(Y_0, Y_1) = \frac{1}{2} \sum_{\alpha \in D^\beta \cap [N], \beta} \left| \Pr[Y_0 = (\alpha; i \in [N], \beta)] - \Pr[Y_1 = (\alpha; i \in [N], \beta)] \right|,
\]

\[
= \frac{1}{2} \sum_{\alpha \in D^\beta \cap [N], \beta \notin D_c(s_{k_{i}}, s_{k_{j}})} \left| \Pr[Y_0 = (\alpha; i \in [N], \beta)] - \Pr[Y_1 = (\alpha; i \in [N], \beta)] \right|, 
\]

\[
+ \frac{1}{2} \sum_{\alpha \in D^\beta \cap [N], \beta \in D_c(s_{k_{i}}, s_{k_{j}})} \left| \Pr[Y_0 = (\alpha; i \in [N], \beta)] - \Pr[Y_1 = (\alpha; i \in [N], \beta)] \right|.
\]

(17)

Step 1. Considering the case of \( \beta \notin D_c(s_{k_{i}}, s_{k_{j}}) \), generally, since

\[
(17) \leq \frac{1}{2} \sum_{\beta \notin D_c(s_{k_{i}}, s_{k_{j}})} \left| \Pr[Y_0 = (\alpha; i \in [N], \beta)] \right| \leq \sum_{\beta \notin D_c(s_{k_{i}}, s_{k_{j}})} \left| \Pr[Y_0^{(N+1)} = \beta] \right|.
\]

(18)

and since for any \( b \in \{0, 1\} \), the variable \( Y_b^{(N+1)} \) is calculated by the hash function \( H(L_{pk}, \mu, \pi_{1}) \), where \( t_{\pi_{1}} = h(\pi_{1}) \), thus \( \beta \notin D_c \) and the probability that \( Y_b^{(N+1)} \) is equal to the given value \( \beta \) is \( 1/|D_c| \). Note that even hash function \( h \in \mathcal{H}(D, D_{x}, m) \) is adversarially chosen, it does not affect the probability that \( Y_0^{(N+1)} \) is equal to a given \( \beta \). Further, for each \( \beta \notin D_c(s_{k_{i}}, s_{k_{j}}) \), \( \Pr[Y_0^{(N+1)} = \beta] = 1/|D_c| \), and since almost all elements of \( D_c \) are in \( D_c(s_{k_{i}}, s_{k_{j}}) \) from Lemma 2, it is evident that the probability of \( \beta \notin D_c(s_{k_{i}}, s_{k_{j}}) \) is negligible. And then we get

(17) \leq \frac{1}{2} \sum_{\beta \notin D_c(s_{k_{i}}, s_{k_{j}})} \left| \Pr[Y_0 = (\alpha; i \in [N], \beta)] \right| = \frac{|D_c| - |D_c(s_{k_{i}}, s_{k_{j}})|}{|D_c|} \geq 1 - \frac{|D_c(s_{k_{i}}, s_{k_{j}})|}{|D_c|} = n^{-o(1)}.

(19)

Step 2. To prove that the value of formula (17) is zero, it is only necessary to prove that each term in formula (17) is zero. Since the last component \( Y_b^{(N+1)} \) of \( Y_b \) is derived from a random oracle \( H \), the probability that \( Y_0^{(N+1)} \) and \( Y_1^{(N+1)} \) are equal to a given value \( \beta \) is the same. Then it is only necessary to prove that the following equation about conditional probabilities holds:

\[
\Pr[Y_b = \alpha, Y_b^{(N+1)} = \beta] = \begin{cases} 
\frac{1}{|D_c|}, & i \in [N] \setminus i_b, \\
\frac{1}{|D_{y}|}, & i = i_b.
\end{cases}
\]

(22)

Therefore, it can be proved that equation (20) holds, i.e., the proof of the lemma is completed.

Suppose that in the game of anonymity in Section 3.2, pp are adversarially chosen according to the Setup algorithm, \( pk_i \) and \( pk_j \) are adversarially generated according to the KeyGen algorithm, message \( \mu \) and ring \( L_{pk} \) are also chosen by adversary. The challenger chooses \( b \in \{0, 1\} \) and invokes
Statistical distance $\Delta_{_{\|\cdot\|}}$ completes the proof.

**Theorem 3** (anonymity). The proposed scheme satisfies UA-CSA, i.e., for the adversary in Definition 10, $X_{b}^{pp,sk_{b},\mu,\ell,\rho_{b}} \rightarrow \text{Sign}(pp, L_{pk}, \mu, sk_{i}), b \in \{0, 1\}$, we have

$$\Delta(X_{0,pp,sk_{0},\mu,\ell,\rho_{0}}, X_{1,pp,sk_{1},\mu,\ell,\rho_{1}}) = n^{-\omega(1)}.$$  \hspace{1cm} (23)

**Proof.** Regarding $X_{b}^{pp,sk_{b},\mu,\ell,\rho_{b}}$, it is known from the signing process, $r_{z,i} \sim D_{z}^{m}$ if $i \neq i_{0}, r_{z,i} = u + c_{i}r$, if $i = i_{0}$. By Lemma 3 and Theorem 2, we know that $r_{z,i}$ is indistinguishable from a randomly selected vector in $D_{z}^{m}$. And since the first component $c_{i}$ in $X_{b}^{pp,sk_{b},\mu,\ell,\rho_{b}}$ is from a random oracle, the statistical distance $\Delta(X_{0,pp,sk_{0},\mu,\ell,\rho_{0}}, X_{1,pp,sk_{1},\mu,\ell,\rho_{1}}) = n^{-\omega(1)}$ for an ordinary adversary $A$. However, for an UA-CSA adversary, obviously, such an analysis is not rigorous enough. The following will focus on the verification process, whether the adversary can distinguish the two signatures based on adversarially-chosen private keys and $c_{i}$ associated with the private key.

Since the first component $c_{i}$ of $X_{b}^{pp,sk_{b},\mu,\ell,\rho_{b}}$ is obtained by accessing a random oracle, and it is not directly related to the associated private key, thus the discussion with respect to $c_{i}$ is not necessary. We only need to prove that for an UA-CSA adversary, the statistical distance between the defined $Y_{0}, Y_{1}$ is negligible. By Lemma 4 it is known that $\Delta(Y_{0}, Y_{1}) = n^{-\omega(1)}$, thus for an UA-CSA adversary, the probability of winning the anonymity game is also negligible. That completes the proof. \hfill $\Box$

5.2. Unforgeability. Before proving unforgeability, the following lemma is first given and proved.

**Lemma 5.** Let $r, r' \sim D_{r}, v, v' \sim D_{v}, c, c' \sim D_{c}$, and $c \neq c'$, if $v - cr = v' - c'r$, then the following inequality holds:

$$v - c'r \neq v' - c'r.$$  \hspace{1cm} (24)

**Proof.** Assume both $v - cr = v' - c'r$ and $v - c'r = v' - c'r$ hold. By subtracting the two equations, we have $(r' - r)(c' - c) = 0$. Since the equation holds in ring $Z_{q}[x]/(x^{n} + 1)$, it does not directly deduce that $r' = r$ or $c' = c$ holds. Due to $\|r'\|_{\lceil q \rceil}, \|c\|_{\lceil q \rceil} \leq n \log n$, $\|c'\|_{\lceil q \rceil}, \|c\|_{\lceil q \rceil} \leq 1$, then the absolute values of the coefficients of $r' - r$ and $c' - c$ are no more than $2n \log n$ and $2$, respectively. Thus $r' - r$ is multiplied by $c' - c$ in ring $Z_{q}[x]/(x^{n} + 1)$, the absolute value of the coefficients of $(r' - r)(c' - c) = 0$ holds in ring $Z_{q}[x]/(x^{n} + 1)$, then it must also hold in ring $Z_{q}[x]/(x^{n} + 1)$. And since $c \neq c'$, then it must have $r' = r$, which is contradictory to the assumption, thus the lemma is proved. \hfill $\Box$

**Theorem 4** (unforgeability). Under the random oracle model, if there exists a polynomial-time adversary $A$ who can validly forge a RS signature about the proposed scheme with probability $\epsilon$, then for a random-chosen $h \in \mathcal{H}(D_{D}, m)$, there is a polynomial-time algorithm $A'$ that can obtain a solution to Col$(h, D_{x})$ with probability at least $\epsilon^{2}/(2^{\sqrt{5}} + 3N\sqrt{\lambda})$, where $N, \varsigma$ and $\psi$ are the number of ring members, the maximum times that $A'$ accesses SO and directly accesses HO, respectively.

**Proof.** Suppose there exists an EU-IC adversary $A'$ who can validly forge a signature about the proposed scheme with non-negligible probability $\epsilon$, then there exists a challenger $A'$ who can solve Col$(h, D_{x})$ with non-negligible probability $\epsilon^{2}$.

Suppose the number of maximum ring members in the system is $l = l(\lambda)$, and given a hash function family $\mathcal{H}(D_{D}, m)$, $A'$ obtains an instance $(h)$ from the Col$(h, D_{x})$ oracle as an input. $A'$ maintains two lists $L_{1}$ and $L_{2}$, which are initialized to be null. For $i \in [l]$, $A'$ honestly runs the Keypgen algorithm to generate the key pair $(pk_{i}, sk_{i})$, and stores the tuple $(i, pk_{i}, sk_{i})$ in list $L_{i}$. $A'$ gives the public key set $S = \{pk_{1}, pk_{2}, \ldots, pk_{N}\}$ to $A''$, and then $A''$ simulates oracles and responds to the queries from $A'$ in the following manner:

Hash query $HO: A''$ submits a set of ring members $L_{pk} = \{pk_{1}, pk_{2}, \ldots, pk_{N}\} \subset S$, message $\mu \in \{0, 1\}^{*}$ and $t_{i} \in D$, $A''$ inquires the list $L_{2}$, and if the tuple $(L_{pk}, \mu, t_{i}, c_{i+1})$ exists, $A''$ returns $c_{i+1}$ to $A'$. Otherwise, $A''$ randomly selects $c_{i+1} \in D_{c}$ and returns it to $A'$, and adds $(L_{pk}, \mu, t_{i}, c_{i+1})$ to the list $L_{i}$.

Signing query $SO$: $A''$ submits an index $\pi$, a message $\mu \in \{0, 1\}^{*}$ and a set of ring members $L_{pk} = \{pk_{1}, pk_{2}, \ldots, pk_{N}\}$ (which may contain some public keys generated by $A'$ in an arbitrary way). Note that since $A'$ knows the private keys of all members in set $S$, and in general, $A''$ will not query signatures about a user outside $S$, which is meaningless, $A'$ responds to this query by honestly running the Sign algorithm.

Corruption query $CO$: $A'$ can make corruption query about any user $U_{i}(i \in [l])$. If $A'$ makes a query on $(i, pk_{i})$, $A'$ first obtains the tuple $(i, pk_{i}, sk_{i})$ by looking for the list $L_{i}$, and then returns $sk_{i}$ to $A'$.

Forgery phase: Suppose that after finishing the above queries, $A''$ outputs a valid forgery $(\mu^{*}, L^{*}_{pk}, \text{sig}^{*})$ with non-negligible probability, and $A''$ did not ask for any signature on $(\ast, \mu^{*}, L^{*}_{pk})$, where $L^{*}_{pk} \subseteq S \setminus C$ and $C$ is the set of corrupted users.

Analysis. Define $p$ as the maximum times $HO$ is queried during $A'$s attack. By Theorem 2, we know that it takes at most $3NH_{O}$ queries to produce a RS, and since $A'$ can make SO queries at most $\xi$ times, the value of $p$ is at most $\psi + 3N\xi$. Suppose that $\text{sig}^{*} = (c_{1}^{*}, (r_{z,i}^{*})_{i \in [N]})$ can pass the verification.
of Verify, then in general, it can be assumed that for some $j$, we have $H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*)) = c^*_{j+1}$.

If $sA$ did not query HO on $(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*))$, then the probability of producing $c^*_{j+1}$ such that $H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*)) = c^*_{j+1}$ is only $1/|D_c|$. Hence, the probability that $c^*_{j+1} \in \{s_1, s_2, \ldots, s_p\}$ is $1 - 1/|D_c|$, where $s_1, s_2, \ldots, s_p \in \{-D_c, \ldots, D_c\}$ is the sequence of return values from HO.

Further, the probability that $sA$ succeeds in a forgery and $c^*_{j+1} \in \{s_1, s_2, \ldots, s_p\}$ is at least $\epsilon - 1/|D_c|$. Suppose $c^*_{j+1} = s_j$, then there are two cases: (1) $s_j$ is a response to $sA$ query (Case 1); (2) $s_j$ is a response to HO query (Case 2). We first analyze Case 1.

**Case 1.** Suppose $c^*_{j+1}$ appears during signing query, and suppose that the corresponding response $sA$ obtains is the signature $s = (c_1, (r_{z,j})_{j \in [N]})$ (i.e., $c_{j+1} = H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*))$). Then compare the signature forged by $sA$ with the legitimate signature $s = (c_1, (r_{z,j})_{j \in [N]})$, we have

$$c^*_{j+1} = H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*)) = H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*)) = c^*_{j+1} = H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*)).$$

(25)

Then either $L^*_{pk} \neq L_{pk}$ or $\mu^* \neq \mu$ (or $r_{z,j}^* \neq r_{z,j}$), since if all three equations hold, this means the adversary simply outputs a signature that has already been asked. If at least one of $L^*_{pk} \neq L_{pk}$ and $\mu^* \neq \mu$ holds, it implies that a collision has occurred in $H$. Hence there is $L^*_{pk} = L_{pk}$, $\mu^* = \mu$, and $r_{z,j}^* \neq r_{z,j}$ with overwhelming probability, that is, $H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*) = c^*_{j+1} = H(L^*_{pk}(\mu^*, h(r_{z,j}^*) - c_j^*pk_j^*)$. Therefore either $h(r_{z,j}^*) - c_j^*pk_j^* \neq h(r_{z,j}) - c_j^*pk_j^*$, which implies there is a collision in $H$, or $h(r_{z,j}^*) - c_j^*pk_j^* \neq h(r_{z,j}) - c_j^*pk_j^*$, which implies there is a collision in $h$. Then we find a collision for $h$, and since $r_{z,j}, r_{z,j}^* \in D_z^* \subseteq D_z$, we find a solution to $Col(h, D_z)$. Note that after $HO$ is queried $p$ times, the probability that a collision happens in $H$ is more than $\rho/|D|$, therefore, the probability of $sA$ outputting a forgery as a solution to $Col(h, D_z)$ is at least

$$\epsilon - \frac{\rho}{|D|}.$$ 

(26)

**Case 2.** The next to consider the case that $s_j$ is obtained by $sA$ calling $HO$. In this scenario, suppose that $sA$ runs $sA$ with some randomness and $sA$ obtains a valid forgery $s = (c_1, (r_{z,j})_{j \in [N]})$ of $\mu^*$. Suppose that $s_1, \ldots, s_p \in \{-D_c, \ldots, D_c\}$ is the sequence of return values from $HO$ during this forgery. Then generate fresh random elements $s^*_1, \ldots, s^*_p \in \{-D_c, \ldots, D_c\}$, and $sA$ runs $sA$ with the same randomness and responds to $sA$’s $HO$ query with $s_1, \ldots, s_{j-1}, s^*_j, \ldots, s^*_p$, by forking lemma from [48], $sA$ will obtain another forged signature $s = (c_1, (r_{z,j})_{j \in [N]})$ on message $\mu^*$ and $s_j \neq s^*_j$ with probability at least $(\epsilon - 1/|D_c|)(\epsilon - 1/|D_c|/\psi + 3|\Lambda| - 1/|D_c|)$. Since the input of $HO$ query is the same in both forgery, we have $h(r_{z,j}^*) - c_j^*pk_j^* = h(r_{z,j}^* - c_j^*pk_j^*)$. By the homomorphic properties of $h$ in Definition 6, it follows that $h(r_{z,j}^* - c_j^*pk_j^*) = h(r_{z,j}^* - c_j^*pk_j^*)$. If $r_{z,j}^* - c_j^*pk_j^* \neq r_{z,j}^* - c_j^*pk_j^*$, and $r_{z,j}^* - c_j^*pk_j^* \neq r_{z,j}^* - c_j^*pk_j^*$, and $r_{z,j}^* - c_j^*pk_j^* \neq r_{z,j}^* - c_j^*pk_j^*$, then the algorithm for finding short vectors on a random lattice by using the well-known lattice reduction algorithm, is $\|z\|_{\infty} \geq \min \left\{ q, 2^\frac{1}{\log\log \rho} \cdot \left( n \log \log \frac{1}{1/4} \right) \right\}$. In the above analysis, we know that the probability of $sA$’ breaking $Col(h, D_z)$ via $sA$ calling $HO$ is at least

$$\left( 1 - \frac{n}{\psi + 3|\Lambda|} \right) \left( 1 - \frac{1}{|D|} \right) \left( 1 - \frac{\rho}{|D|} \right).$$

(27)

By comparing formulas (26) and (27), we have the inequality (26) > (27). Thus, the probability of $sA$ running $sA$ to solve $Col(h, D_z)$ successfully is

$$\epsilon^* \geq \left( 1 - \frac{n}{\psi + 3|\Lambda|} \right) \left( 1 - \frac{1}{|D|} \right) \left( 1 - \frac{\rho}{|D|} \right).$$

(28)

**6. Discussion**

From Theorem 4, it is known that if an adversary succeeds in forging an RS against the proposed scheme, then the challenger can find a collision for a randomly chosen hash function $h_0(\ast)$ in $\mathcal{H}(D, D_x, m)$. From [29], solving $Col(h, D_z)$ is equivalent to finding a vector $u \in \Lambda^*_n(A)$ on lattice $\Lambda^*_n(A) = \{u \in \mathbb{Z}^m \mid \text{Au} = 0 \mod q\}$ satisfying $\|u\|_{\infty} \leq 2(mn)^{1/2} \log n + n \log^2 n$ where $A = [\text{Rot}(a_1)] \ldots [\text{Rot}(a_n)] \in \mathbb{Z}^{m \times m}$, $a_i$ is the $i$-th component of $a$, $i \in [m]$. It was shown in [49] that in a reasonable amount of time, the algorithm for finding short vectors on a random lattice will generate a vector no less than $1.01^n$ times the shortest vector over the lattice. Furthermore, based on the experiments of [49], Micciancio and Regev [50] conduct experiments on lattices very similar to $\Lambda^*_n(A)$, and proves that the length of the shortest vector, which can be found on $\Lambda^*_n(A)$ by using the well-known lattice reduction algorithm, is $\|z\|_{\infty} \geq \min \left\{ q, 2^\frac{1}{\log\log \rho} \cdot \left( n \log \log \frac{1}{1/4} \right) \right\}$. And the corresponding $\|z\|_{\infty}$ at least,
Table 1: Parameter setting for our scheme.

| Parameter | n     | q (≈) | m  | The vector length needed to break RS | The shortest vector length that can be found | Security level (bits) |
|-----------|-------|-------|----|-------------------------------------|-------------------------------------------|----------------------|
| Recommended choice | 128   | 2^79  | 16 | 2^18.4                              | 2^19.2                                    | 80                   |
|           | 256   | 2^43  | 8  | 2^19.1                              | 2^20.3                                    | 80                   |
|           | 512   | 2^41  | 5  | 2^20.1                              | 2^29.6                                    | 128                  |
|           | 1024  | 2^73  | 8  | 2^22.4                              | 2^59.9                                    | 192                  |

Table 2: Comparison of storage overhead and security.

| Scheme       | Public key size | Private key size | Signature size | UA- CSA | EU-IC |
|--------------|-----------------|------------------|----------------|---------|-------|
| [37]         | O(n log q)      | O(n log n)       | O((log (Nn)n log² n)) | Yes     | Yes   |
| [38]         | O(n log q)      | O(n)             | O(2 log² Nn log n) | Yes     | Yes   |
| [40]         | O(n log q)      | O(n)             | O((log Nn log(nq)) | Yes     | Yes   |
| AM1 of [41]  | O(n log n log q) | O(n log n)       | O(Nn log n log (n^1.5 log^5 n)) | Yes | No    |
| AM2 of [41]  | O(λn log n log q) | O(λ log n)       | O(Nn log n log (n^1.5 log^5 n)) | Yes | Yes   |
| Ours         | O(n log (√n log n)) | O(n log (√n log n)) | O(Nn log n log (n^1.5 log^5 n)) | Yes | Yes   |

Table 3: Comparison of computational overhead at security level λ = 128.

| Scheme       | Key generation cost | Signature cost | Verification cost |
|--------------|---------------------|----------------|-------------------|
| [38]         | 438T_{poly-mul}      | 60N + 9660T_{poly-mul} | 6N + 967T_{poly-mul} |
| AM1 of [41]  | 21T_{poly-mul} + T_{poly-inv} | 63NT_{poly-mul} | 21NT_{poly-mul} |
| AM2 of [41]  | λ(21T_{poly-mul} + T_{poly-inv})/2 | 63NT_{poly-mul} + T_v + T_c | 21NT_{poly-mul} |
| Ours         | 5T_{poly-mul}        | 15NT_{poly-mul} | 5NT_{poly-mul}    |

Table 4: Comparison of approximate computational costs under ring size N = 2^7.

| Scheme       | Key generation cost | Signature cost | Verification cost |
|--------------|---------------------|----------------|-------------------|
| [38]         | 438T_{poly-mul}      | 17340T_{poly-mul} | 8647T_{poly-mul}  |
| AM1 of [41]  | 21T_{poly-mul} + T_{poly-inv} | 8064T_{poly-mul} | 26887T_{poly-mul} |
| AM2 of [41]  | λ(21T_{poly-mul} + T_{poly-inv})/2 | 8064T_{poly-mul} + T_v + T_c | 26887T_{poly-mul} |
| Ours         | 5T_{poly-mul}        | 1920T_{poly-mul} | 6407T_{poly-mul}  |

Therefore, in order to make Col (h, D_x) be intractable, we should set the parameters in such a manner that requires 2(mn^{1.5} log n + n log^2 n) to be smaller than formula (29). For example, n = 512, q = 2^{41}, then calculate 2(mn^{1.5} log n + n log^2 n) = 2^{220.1}, while the infinite norm of the shortest vector ∥z∥_∞ that can be found by formula (29) is around 2^{29.6}. Table 1 gives several sets of parameter settings and security levels for our scheme.

7. Efficiency

In this section, in respect of efficiency and security, we compare our scheme with five lattice-based RS schemes without trapdoors: the RS schemes of [37, 38, 40] and the two RS schemes AM1 and AM2 of [41]. The comparison results of storage overhead and computational overhead are listed in Table 2 and 3, respectively. λ is a security parameter, n > λ is a power of 2, N is the number of ring members, and q is a large prime number. In estimating the computational efficiency of each scheme, we mainly focus on the relatively time-consuming operations, such as polynomial inversion and polynomial-polynomial multiplication, while ignoring the less time-consuming operations such as polynomial-polynomial addition and cryptographic hash operation H.

The notation T_{poly-inv} denotes the computational overhead of running polynomial-polynomial multiplication in D once, T_{poly-mul} denotes the computational overhead of running polynomial inversion once, normally T_{poly-inv} > T_{poly-mul}. T_v denotes the time spent in performing λ times sanity checks of parameters in the signing phase of AM2. T_c denotes the computational overhead of selecting a compliant signing key in the signing phase of AM2.

From Table 2, the signature size of [37, 38, 40] is O((log N)N), while the signature size of AM1, AM2 and ours is O(N). Without loss of generality, n ≥ √ n log n, thus compared with AM1, AM2, our scheme has smaller public/private key size and signature size.

These schemes [37, 38, 40] all use complex and bloated zero-knowledge proofs, and it is very difficult for us to calculate their computational overheads. Among these schemes [37, 38, 40], only the relatively concise scheme of [38] gives its computational overheads analysis. Therefore, we only compare the computational cost of [38, 41] and our scheme. We adopt the calculation method of efficiency in
[38], and reduce the operations of KeyGen, Sign and Verify to polynomial-polynomial multiplication operations. To make the comparison results more intuitive, the parameters of [38, 41] and ours are set under the same security level $\lambda = 128$. Here, $n$ in [38], AM, and ours are set to 256, 128, and 512, respectively; $m$ in [38], AM, and ours are set to 15, 21, and 5, respectively. In terms of KeyGen, Sign and Verify, our scheme obviously has the smallest computational cost as shown in Tables 3 and 4.

To make the comparison results more intuitive, Figures 1 and 2 show the comparison between [38, 41] and ours in terms of computational overhead under different ring sizes.

8. Conclusions

Although lattice-based RS schemes are resistant to attacks by quantum computers, there still exists a big gap between them and the schemes that are based on traditional number-theoretic assumptions with respect to computational efficiency. To further boost the computational efficiency of RS on lattice, this work constructs a lattice-based RS scheme without trapdoors under the random oracle model. Based on a collision-resistant hash function on ideal lattice, our scheme is designed via the FSwA protocol. Our scheme avoids the use of trapdoors or sampling techniques that have high computational overhead and does not involve complex zero-knowledge proofs which are usually used in RS schemes without trapdoors. The proposed scheme is more concise and efficient. Meanwhile, in terms of anonymity and unforgeability, our scheme is proven to satisfy the strongest UA-CSA and EU-IC, respectively. Next, we plan to investigate NTRU lattice-based RS scheme without trapdoors, which will be the first combination of NTRU lattice and RS without trapdoors.

Data Availability

The figures and tables used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61802117), the Support Plan of Scientific and Technological Innovation Team in Universities of Henan Province (20IRTSTHN013), and the Youth Backbone Teacher Support Program of Henan Polytechnic University (2018XQG-10).

References

[1] R. L. Rivest, A. Shamir, and Y. Tauman, “How to leak a secret,” in Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, pp. 552–565, Beijing, China, December, 2001.
[2] M. Abe, M. Ohkubo, and K. Suzuki, “1-out-of-n signatures from a variety of keys,” in Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security, pp. 415–432, Jeju Island, Korea, December, 2002.
[3] E. Bresson, J. Stern, and M. Szydlo, “Threshold ring signatures and applications to ad-hoc groups,” in Proceedings of the 22nd Annual International Cryptology Conference, pp. 465–480, California, USA, August, 2002.
[4] F. G. Zhang and K. Kim, “Id-based blind signature and ring signature from pairings,” in Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security, pp. 533–547, Queenstown, New York, December, 2002.
[5] M. Naor, “Deniable ring authentication,” in Proceedings of the 22nd Annual International Cryptology Conference, pp. 481–498, Santa Barbara, CA, USA, August, 2002.
[6] D. Boneh, C. Gentry, and B. Lynn, “Aggregate and verifiably encrypted signatures from bilinear maps,” in Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, pp. 416–432, Zagreb, Croatia, May, 2003.
[7] J. Herranz and G. Saez, “Forking lemmas for ring signature schemes,” in Proceedings of the 4th International Conference on Cryptology in India, pp. 266–279, Jaipur, India, December, 2003.
[8] J. K. Liu, V. K. Wei, and D. S. Wong, “Linkable spontaneous anonymous group signature for ad hoc groups,” in Proceedings of the 9th Australasian Conference on Information Security and Privacy, pp. 325–335, Perth Australia, January, 2004.
[9] B. Adida, S. Hohenberger, and R. L. Rivest, “Ad-hoc-group signatures from hijacked keypairs,” in DIMACS Workshop on Theft in E-Commerce, Piscataway, NJ, USA, 2005, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.4165&rep=rep1&type=pdf.

[10] J. Xu, Z. Zhang, and D. Feng, “A Ring Signature Scheme Using Bilinear Pairings,” in International Workshop on Information Security Applications, vol. 3325, pp. 160–169, 2004.

[11] A. Bender, J. Katz, and R. Morselli, “Ring signatures: stronger definitions, and constructions without random oracles,” Journal of Cryptology, vol. 22, no. 1, pp. 114–138, 2009.

[12] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM Review, vol. 41, no. 2, pp. 303–332, 1999.

[13] M. Ajtai, “Generating hard instances of lattice problems,” in Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 99–108, Pennsylvania USA, May, 1996.

[14] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” Journal of the ACM, vol. 56, no. 6, pp. 1–40, 2009.

[15] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” in Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 1–23, French Riviera, France, May, 2010.

[16] R. Lindner and C. Peikert, “Better Key Sizes (And Attacks) for LWE-Based Encryption,” Cryptographers’ Track at the RSA Conference, pp. 319–339, Springer, San Francisco, CA, USA, 2011.

[17] W. Susilo, P. Dutta, and D. H. Duong, “Lattice-based HRA-secure attribute-based proxy re-encryption in standard model,” in Proceedings of the 26th European Symposium on Research in Computer Security, pp. 169–191, Darmstadt, Germany, October, 2021.

[18] J. M. B. Mera, A. Karmakar, and T. Marc, “Efficient lattice-based inner-product functional encryption,” in Proceedings of the 25th IACR International Conference on Public-Key Cryptography, pp. 163–193, Virtual Event, August, 2022.

[19] O. Goldreich, S. Goldwasser, and S. Halevi, “Public-key cryptosystems from lattice reduction problems,” in Proceedings of the 17th Annual International Cryptology Conference, pp. 112–131, Berlin, Heidelberg, August, 1997.

[20] J. Hofstein, J. Pipher, and J. H. Silverman, “ANTRU lattice-based signature scheme,” in Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 211–228, Russia, May, 2001.

[21] J. Hofstein, N. Howgrave-Graham, and J. Pipher, “NTRU-Sign: Digital Signatures Using the NTRU Lattice,” Cryptographers’ Track at the RSA Conference, pp. 122–140, Springer, San Francisco, CA, USA, 2003.

[22] Y. P. Hu, B. C. Wang, and W. C. He, “NTRUSign with a new perturbation,” IEEE Transactions on Information Theory, vol. 54, no. 7, pp. 3216–3221, 2008.

[23] P. Q. Nguyen and O. Regev, “Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures,” Journal of Cryptology, vol. 22, no. 2, pp. 139–160, 2009.

[24] L. Lucas and P. Q. Nguyen, “Learning a zonotope and more: cryptanalysis of NTRUSign countermeasures,” in Proceedings of the 18th Annual International Conference on the Theory and Application of Cryptology and Information Security, pp. 433–450, Tokyo, Japan, April, 2012.

[25] C. Gentry, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” in Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 197–206, New York, USA, June, 2008.

[26] D. Micciancio and C. Peikert, “Trapdoors for lattices: simpler, tighter, faster, smaller,” in Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 700–718, Cambridge, UK, April, 2012.

[27] J. Alwen and C. Peikert, “Generating shorter bases for hard random lattices,” Theory of Computing Systems, vol. 48, no. 3, pp. 535–553, 2011.

[28] L. Ducas and D. Micciancio, “Improved short lattice signatures in the standard model,” in Proceedings of the 34th Annual Cryptology Conference, pp. 335–352, Santa Barbara, CA, USA, April, 2014.

[29] V. Lyubashevsky, “Fiat-Shamir with aborts: applications to lattice and factoring-based signatures,” in Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, pp. 598–616, Tokyo, Japan, August, 2009.

[30] V. Lyubashevsky, “Lattice signatures without trapdoors,” in Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 738–755, Cambridge, UK, April, 2012.

[31] J. Brakerski and Y. T. Kalai, “A framework for efficient signatures, ring signatures and identity based encryption in the standard model,” 2010, http://eprint.iacr.org/2010/086.

[32] J. Wang and B. Sun, “Ring signature schemes from lattice basis delegation,” in Proceedings of the 13th International Conference on Information and Communications Security, pp. 15–28, Beijing China, November, 2011.

[33] Y. H. Zhang, Y. P. Hu, J. Xie, and M. Jiang, “Efficient ring signature schemes over NTRU lattices,” Security and Communication Networks, vol. 9, no. 18, pp. 5252–5261, 2016.

[34] G. M. Zhao and M. M. Tian, “A simpler construction of identity-based ring signatures from lattices,” in Proceedings of the 12th International Conference on Provably Secure, pp. 277–291, South Korea, October, 2018.

[35] X. Y. Lu, M. H. Au, and Z. F. Zhang, “Raptor: a practical lattice-based (linkable) ring signature,” in Proceedings of the 17th Applied Cryptography and Network Security, pp. 110–130, Bogota, Colombia, June, 2019.

[36] Y. L. Tang, F. F. Xia, and Q. Ye, “Identity-based linkable ring signature on NTRU lattice,” Security and Communication Networks, vol. 2021, Article ID 9992414, 17 pages, 2021.

[37] S. Libert, S. Ling, and K. Nguyen, “Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors,” in Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 1–31, Vienna, Austria, May, 2016.

[38] M. F. Esgin, R. Steinfeld, and J. K. Liu, “Lattice-based zero-knowledge proofs: new techniques for shorter and faster constructions and applications,” in Proceedings of the 39th Annual International Cryptology Conference, pp. 115–146, Santa Barbara, CA, USA, August, 2019.

[39] W. Beullens, S. Katsumata, and F. Pintore, “Calamari and factoring-based cryptography: new techniques for shorter and faster constructions and applications,” in Proceedings of the 34th Annual International Conference on the Theory and Application of Cryptography and Information Security, pp. 464–492, Daejeon, November, 2020.

[40] V. Lyubashevsky, N. K. Nguyen, and G. Seiler, “SMILE: set membership from ideal lattices with applications to ring signatures and confidential transactions,” in Proceedings of the
41st Annual International Cryptology Conference, pp. 611–640, Virtual Event, August, 2021.

[41] C. Aguilar Melchor, S. Bettaieb, and X. Boyen, “Adapting Lyubashevsky’s signature schemes to the ring signature setting,” in Proceedings of the 6th International Conference on Cryptology in Africa, pp. 1–25, Cairo, Egypt, July, 2013.

[42] W. A. Alberto Torres, R. Steinfeld, and A. Sakzad, “Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1.0),” in Proceedings of the 23rd Australasian Conference on Information Security and Privacy, pp. 558–576, Perth, WA, Australia, March, 2018.

[43] C. Baum, H. Lin, and S. Oechsner, “Towards practical lattice-based one-time linkable ring signatures,” in Proceedings of the 20th International Conference on Information and Communications Security, pp. 303–322, Lille, France, August, 2018.

[44] Z. Liu, K. Nguyen, and G. Yang, “A lattice-based linkable ring signature supporting stealth addresses,” in Proceedings of the 24th European Symposium on Research in Computer Security, pp. 726–746, Luxembourg, September, 2019.

[45] V. Lyubashevsky and D. Micciancio, “Generalized compact knapsacks are collision resistant,” in Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming, pp. 144–155, Venice Italy, July, 2006.

[46] D. Micciancio, “Improved cryptographic hash functions with worst-case/average-case connection,” in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pp. 609–618, Dallas Texas USA, May, 2002.

[47] V. Lyubashevsky, Towards practical lattice-based cryptography, PhD Thesis, University of California, San Diego, 2008.

[48] M. Bellare and G. Neven, “Multi-signatures in the plain public-key model and a general forking lemma,” in Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 390–399, Berlin, Germany, May, 2006.

[49] N. Gama and P. Q. Nguyen, “Predicting lattice reduction,” in Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 31–51, Istanbul, Turkey, April, 2008.

[50] D. Micciancio and O. Regev, “Lattice-based cryptography,” Post-Quantum Cryptography, pp. 147–191, 2009.