Lattice-Based Identity-Based Linkable Ring Signature Scheme Postprint

Authors: Liu Mengqing, Wang Xueming

Date: 2022-05-18T16:08:25+00:00

Abstract

To resist quantum algorithm attacks and address the vulnerability where malicious signers exploit the complete anonymity of ring signature technology to output multiple signatures for double-spending attacks, while simultaneously solving the problem of unnecessary system overhead waste, a novel lattice-based identity-based linkable ring signature scheme is proposed. The scheme is founded on the security of the approximate shortest vector problem on lattices, reduces the solution of this problem to that of a collision problem, employs linear operations between matrices and vectors to generate signatures, and integrates identity-based cryptographic techniques. It resolves the system overhead waste issue, avoids complex algorithms such as trapdoor generation and Gaussian sampling, enhances signature efficiency, reduces storage overhead, and is proven under the random oracle model to satisfy complete anonymity and strong existential unforgeability. Analysis demonstrates that this scheme constitutes a secure and efficient ring signature solution.

Full Text

Identity-based Linkable Ring Signature Scheme from Lattices

Liu Mengqing[1],[2], **Wang Xueming**[2]† [1]State Key Laboratory of Public Big Data, [2]College of Computer Science & Technology, Guizhou University, Guiyang 550025, China

Abstract: To resist quantum algorithm attacks and address the vulnerability where malicious signers exploit the complete anonymity of ring signature technology to output multiple signatures for double-spending attacks, while simultaneously solving the problem of unnecessary system overhead waste, this paper proposes a novel identity-based linkable ring signature scheme from lattices. The scheme is based on the approximate shortest vector problem (SVP)

on lattices, reducing its solution to the solution of a collision problem, and generates signatures using linear operations between matrix vectors while incorporating identity-based cryptography. It solves the problem of system overhead waste, avoids complex algorithms such as trapdoor generation and Gaussian sampling, improves signature efficiency, and reduces storage overhead. The scheme is proven to satisfy complete anonymity and strong existential unforgeability under the random oracle model. Analysis demonstrates that this is a secure and efficient ring signature scheme.

Keywords: linkable ring signature; lattice; identity-based cryptography; random oracle model

0 Introduction

Electronic transactions have become an unstoppable trend, with digital signatures [**?**] playing an indispensable role. However, the privacy of signers' identities cannot be guaranteed in ordinary digital signatures. To address this issue, ring signatures [**?**] were proposed and have since been developed. Nevertheless, in blockchain mechanisms, the strong anonymity of ring signatures allows malicious signers to output two or more different signatures for the same event, thereby suffering from double-spending attacks [**?**]. Consequently, Liu et al. [**?**] proposed linkable ring signatures (LRS) in 2004. Linkability can detect whether two signatures were signed by the same user. Currently, LRS has found many applications, such as in e-commerce activities [**?**, **?**, **?**, **?**, **?**]. Identity-based cryptography can reduce wasteful system overhead, and the first identity-based linkable ring signature (IBLRS) [**?**] was proposed in 2006. Early proposed schemes [**?**] were later proven to have security flaws. Most traditional ring signature schemes currently face the risk of being broken after the emergence of quantum algorithms, making research on post-quantum cryptography a frontier direction in cryptography. Among various post-quantum cryptographic technologies, lattice-based cryptography stands out due to its inherent advantages and has become the most prominent post-quantum cryptographic technology.

Lattice-based cryptography was proposed by Ajtai [**?**]. Solving lattice hard assumptions remains difficult even for quantum computers [**?**]. In 2008, Gentry et al. [**?**] proposed a "hash-and-sign" signature scheme based on hard lattice problems, a mechanism widely applied [**?**, **?**, **?**, **?**]. However, the "hash-and-sign" signature mechanism suffers from high storage overhead and computational inefficiency. Apart from constructing lattice-based ring signatures using the "hash-and-sign" mechanism, Lyubashevsky [**?**] presented a signature scheme based on the approximate shortest vector problem using the Fiat-Shamir transform, and additionally proposed rejection sampling technology [**?**]. In 2018, Torres et al. [**?**] developed the first post-quantum one-time linkable ring signature. This scheme, based on the Ring-SIS hard assumption on lattices and employing rejection sampling technology, improved the independence of signature private

keys and was applied in blockchain transactions. In the same year, Baum et al. [**?**] proposed a simpler and more efficient lattice-based LRS scheme. In 2019, Torres et al. [**?**] extended the scheme from [**?**], resulting in version 2.0 of the LRS scheme, which used lattice-based zero-knowledge proofs to achieve security against out-of-range attacks. In 2021, Tang Yongli et al. [**?**] proposed an IBLRS scheme employing trapdoor generation and preimage sampling algorithms. This scheme uses computationally complex algorithms that increase time overhead, while the large trapdoor size also increases storage overhead.

This paper constructs a new lattice-based identity-based linkable ring signature (IBLRS) scheme. The scheme reduces the solution of the approximate shortest vector problem on lattices to the solution of a collision problem, does not use Gaussian sampling or trapdoor techniques, and all computations are based on simple multiplication operations between matrix vectors, enabling higher computational efficiency. The scheme' s security is verified under the random oracle model, and its efficiency is analyzed.

---

1 Preliminaries

1.1 Notation

Table 1 provides a brief explanation of symbols used in this paper.

Table 1: Notations

Symbol	Description
$\mathbb{R}$ ($\mathbb{Z}$)	Set of real numbers (integers)
2^{κ}	Power of 2
$f(x)$	Irreducible polynomial
$\mathbf{a}$	Polynomial vector
D	Uniform random sample from $\mathbb{R}$
$\mathcal{H}$	Hash function family

1.2 Lattice Theory

Definition 1 (Lattice). Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\} \subset \mathbb{R}^n$ constitute a matrix where $\mathbf{b}_i$ are m linearly independent vectors. The lattice Λ generated by $\mathbf{B}$ refers to the linear combination set with integer coefficients, i.e.:

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{m} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}$$

Let Λ be a q-cyclic lattice [**?**]. If for any $\mathbf{v} \in \Lambda$, the cyclic shift of $\mathbf{v}$ is also in Λ.

Definition 2 (Shortest Vector Problem, SVP). The goal of SVP is to find a non-zero vector with the smallest Euclidean norm in a given arbitrary lattice Λ. Simply put, in any given lattice Λ, for any lattice vector $\mathbf{v} \in \Lambda$, there exists a non-zero vector $\mathbf{u} \in \Lambda$ such that $\|\mathbf{u}\| \leq \|\mathbf{v}\|$.

Definition 3 (Approximate Shortest Vector Problem, SVP_γ). Given an n-dimensional arbitrary lattice Λ, the goal of the approximate shortest vector problem is to find a non-zero vector $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\Lambda)$ holds, where $\lambda_1(\Lambda)$ is the length of the shortest non-zero vector in Λ and $\gamma \geq 1$ is a rational number.

1.3 Collision-Resistant Hash Function Family

Definition 4 [?] (Collision-Resistant Hash Function Family). For $m > n \log q$ and integer $q > 2$, let $\mathcal{H} = \{h : \mathbb{R}^m \to \mathbb{R}^n\}$ be a hash function family. The collision problem $\mathrm{Col}(h, D)$ for $h \in \mathcal{H}$ and a given domain $D \subset \mathbb{R}^m$ aims to find two different vectors $\hat{\mathbf{s}}, \hat{\mathbf{s}}' \in D$ satisfying $h(\hat{\mathbf{s}}) = h(\hat{\mathbf{s}}')$.

For any q-cyclic lattice, the collision problem $\mathrm{Col}(h, D)$ is as hard as solving SVP_γ on the q-cyclic lattice.

Theorem 1 [?]. Given a random hash function $h \in \mathcal{H}$ as described above, if there exists an algorithm that can break the collision problem $\mathrm{Col}(h, D)$ with non-negligible probability, then there must exist an algorithm that can solve SVP_γ on any q-cyclic lattice with non-negligible probability.

1.4 Statistical Distance

Definition 5 (Negligible Function). If for any integer c and sufficiently large n, $f(n) < n^{-c}$ holds, then function $f(n)$ is negligible. Negligible functions are typically denoted by $\mathrm{negl}(n)$.

Definition 6 (Statistical Distance). Let X and Y be two random variables in finite domain S. The statistical distance between X and Y is defined as:

$$\Delta(X, Y) = \frac{1}{2} \sum_{x \in S} |\Pr[X = x] - \Pr[Y = x]|$$

If the statistical distance between X and Y is $\Delta(X, Y) < \mathrm{negl}(n)$, then X and Y are statistically indistinguishable.

---

2 Security Model

2.1 General Definitions

An identity-based linkable ring signature scheme generally consists of five polynomial-time algorithms: Setup, Extract, RingSign, Verify, and Link.

Setup(n): A randomized algorithm executed by the Key Generation Center (KGC). It takes security parameter n as input and outputs public parameters PP and master secret key MSK.

Extract(PP, ID, MSK): A randomized algorithm executed by the KGC. It takes PP, user identity information ID, and MSK as input, and outputs the private key SK_{ID} corresponding to user ID.

RingSign$(PP, \mu, ID, U, SK_{ID})$: A randomized algorithm executed by the signing user. It takes PP, message μ to be signed, ring U, and corresponding SK_{ID} as input, and outputs ring signature Sig for μ.

Verify(PP, U, μ, Sig): A deterministic algorithm executed by the verifying user. It takes public parameters PP, ring U, message μ, and corresponding ring signature Sig as input. It returns "1" if verification passes and "0" otherwise.

Link(Sig_1, Sig_2): Executed by the verifier, this algorithm takes two message-signature pairs (μ_1, Sig_1) and (μ_2, Sig_2) as input. If they were generated by the same signer for the same event, it outputs "link"; otherwise, it outputs "unlink".

Correctness: The correctness of a linkable ring signature includes signature correctness and link correctness.

Signature correctness means that if Sig is a legitimate signature output by RingSign, the probability that verification algorithm Verify outputs "0" is negligible:

$$\Pr\left[\text{Verify}(PP, U, \mu, \text{RingSign}(PP, \mu, ID, U, SK_{ID})) = 0\right] \leq \text{negl}(n)$$

Link correctness means that for tuples (μ_1, Sig_1) and (μ_2, Sig_2), if they were generated by the same signer for the same event, the probability that link algorithm Link outputs "unlink" is negligible:

$$\Pr\left[\text{Link}(Sig_1, Sig_2) = \text{"unlink"} \mid \text{same signer}\right] \leq \text{negl}(n)$$

2.2 Security Model

A secure IBLRS scheme must satisfy the following properties [**?**]:

Definition 7 (Anonymity). Consider the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$:

Setup: Input n, execute Setup algorithm to obtain PP, MPK, and MSK, and send PP and MPK to $\mathcal{A}$.

Query: $\mathcal{A}$ can make the following queries: - **Extract query**: $\mathcal{A}$ submits a user identity ID to $\mathcal{C}$, who runs Extract and returns the corresponding private key SK_{ID}. - **Sign query**: $\mathcal{A}$ submits a ring U, message μ, and user identity ID to $\mathcal{C}$, who calls RingSign to sign the message and returns signature Sig.

Challenge: After completing queries, $\mathcal{A}$ submits a message μ^*, ring U^*, and two user identities ID_0 and ID_1 (where $ID_0, ID_1 \in U^*$). Challenger $\mathcal{C}$ randomly selects $b \in \{0,1\}$, runs RingSign with the private key corresponding to ID_b to sign message μ^* and ring U^*, and outputs a ring signature Sig^* to $\mathcal{A}$.

Guess: $\mathcal{A}$ outputs a guess b' of the random bit b. If $b' = b$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ in the above game is defined as:

$$\mathbf{Adv}_{\mathcal{A}}^{\text{anon}}(n) = |\Pr[b' = b] - \frac{1}{2}|$$

If this advantage is negligible for any polynomial-time adversary $\mathcal{A}$, the identity-based linkable ring signature scheme satisfies anonymity.

Definition 8 (Strong Existential Unforgeability). Consider the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$:

Setup: Input n, run Setup algorithm to obtain PP, MPK, and MSK, and send PP and MPK to adversary $\mathcal{A}$.

Query: $\mathcal{A}$ can make the following queries: - **Hash query**: Adversary $\mathcal{A}$ submits a message μ and ring U to challenger $\mathcal{C}$, who returns the corresponding hash value. - **Extract query**: Adversary $\mathcal{A}$ submits user identity ID to challenger $\mathcal{C}$, who runs Extract and returns the corresponding private key SK_{ID}. - **Sign query**: Adversary $\mathcal{A}$ submits a ring U, user identity ID, and message μ to challenger $\mathcal{C}$, who calls RingSign to sign the message and returns signature Sig.

Forge: Adversary $\mathcal{A}$ outputs a tuple (U^*, μ^*, Sig^*). $\mathcal{A}$ successfully forges a signature and wins the game if: a) $\mathcal{A}$ never initiated a signature query for (U^*, μ^*); b) The private key of any member in U^* was not queried; c) $\text{Verify}(PP, U^*, \mu^*, Sig^*) = 1$.

The advantage of $\mathcal{A}$ in the above game is defined as:

$$\mathbf{Adv}_{\mathcal{A}}^{\text{forge}}(n) = \Pr[\mathcal{A} \text{ wins the game}]$$

If this advantage is negligible for any adversary $\mathcal{A}$, the identity-based linkable ring signature scheme satisfies strong existential unforgeability.

Definition 9 (Linkability). Consider the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$:

Setup: Input n, run Setup algorithm to obtain PP, MPK, and MSK, and send PP and MPK to adversary $\mathcal{A}$.

Query: $\mathcal{A}$ can make the following queries: - **Hash query**: Adversary $\mathcal{A}$ submits a message μ and ring U to challenger $\mathcal{C}$, who returns the corresponding hash value. - **Extract query**: Adversary $\mathcal{A}$ submits user identity ID to challenger $\mathcal{C}$, who runs Extract and returns the corresponding private key SK_{ID}. -

Sign query: Adversary $\mathcal{A}$ submits a message μ, ring U, and user identity ID to challenger $\mathcal{C}$, who calls RingSign to sign the message and returns signature Sig.

Forge: Finally, adversary $\mathcal{A}$ outputs two ring signatures $(U_1^*, \mu_1^*, Sig_1^*)$ and $(U_2^*, \mu_2^*, Sig_2^*)$ that satisfy the following conditions: a) $\mathcal{A}$ never initiated a signature query for either (U_1^*, μ_1^*) or (U_2^*, μ_2^*); b) The private keys of any members in U_1^* and U_2^* were not queried; c) $\mathcal{A}$ possesses at most one user' s private key; d) Verify$(PP, U_i^*, \mu_i^*, Sig_i^*) = 1$ for $i \in \{1, 2\}$; e) Link$(Sig_1^*, Sig_2^*) =$ "unlink".

If $\mathcal{A}$ outputs such signatures, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ in the above game is defined as:

$$\mathbf{Adv}_{\mathcal{A}}^{\text{link}}(n) = \Pr[\mathcal{A} \text{ wins the game}]$$

If this advantage is negligible for any adversary $\mathcal{A}$, the identity-based linkable ring signature scheme satisfies linkability.

---

3 Lattice-Based Identity-Based Linkable Ring Signature Scheme

This section constructs the lattice-based IBLRS scheme and analyzes it. Before constructing the scheme, we explain some variables as shown in Table 2.

Table 2: Variables

Symbol	Description
n	Security parameter
m	Dimension parameter
q	Prime modulus
D	Domain of vectors
R	Range of hash functions
$\mathcal{H}$	Hash function family
h	Hash function
ID	User identity
SK_{ID}	Private key of identity ID
U	Ring (set of identities)
μ	Message
Sig	Signature

3.1 Scheme Construction

Setup(n): Given security parameter n (where n is a power of 2), determine a maximum ring user set $U_{\max}$, where max denotes the maximum number of users.

Choose a prime p such that when $n > 4$, the inequality $\log p > 1.54 \log d + \log n$ holds. Randomly select a hash function h from the collision-resistant hash function family $\mathcal{H}$. Choose a random oracle function H. Randomly select $\hat{\mathbf{s}} \in D$, compute $\mathbf{S} = h(\hat{\mathbf{s}})$. Set $PP = \{n, m, p, D, R, \mathcal{H}, h, H\}$, $MPK = \mathbf{S}$, and $MSK = \hat{\mathbf{s}}$.

Extract(PP, ID, MSK): Input PP, MSK, and user identity $ID \in \{0,1\}^*$. Perform the following computation: 1. Compute $\hat{\mathbf{r}}_{ID} = H(ID)$ and $Q_{ID} = h(\hat{\mathbf{r}}_{ID})$. 2. The private key for user identity ID is $SK_{ID} = \hat{\mathbf{s}} + \hat{\mathbf{r}}_{ID}$.

RingSign$(PP, \mu, ID, U, SK_{ID})$: A randomized algorithm executed by the signing user. Input PP, message μ to be signed, ring $U = \{ID_1, ID_2, \ldots, ID_l\}$, and the signer' s private key SK_{ID}. The signing process is as follows: 1. For each $j \in [l]$, compute $c_j = H(\mu, U, ID_j)$. 2. For the actual signer with identity ID_i, randomly select $\hat{\mathbf{y}}_i \in D$ and compute $\mathbf{Y}_i = h(\hat{\mathbf{y}}_i)$. 3. For $j \neq i$, randomly select $\hat{\mathbf{z}}_j \in D$. 4. Compute the link tag $\hat{\mathbf{I}} = H(SK_{ID})$. 5. For $j = i$, compute $\hat{\mathbf{z}}_i = c_i \cdot SK_{ID} + \hat{\mathbf{y}}_i$. 6. Output signature $Sig = (\hat{\mathbf{I}}, \hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots, \hat{\mathbf{z}}_l)$.

Verify(PP, U, μ, Sig): Given PP, ring $U = \{ID_1, ID_2, \ldots, ID_l\}$, message μ, and signature $Sig = (\hat{\mathbf{I}}, \hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_l)$. For each $j \in [l]$, compute $c_j = H(\mu, U, ID_j)$ and verify whether:

$$h(\hat{\mathbf{z}}_j) = c_j \cdot (MPK + H(ID_j)) + h(\hat{\mathbf{I}})$$

Return "1" if all equations hold; otherwise, return "0" .

Link(Sig_1, Sig_2): Input two ring signatures $Sig_1 = (\hat{\mathbf{I}}_1, \hat{\mathbf{z}}_{1,1}, \ldots, \hat{\mathbf{z}}_{1,l_1})$ and $Sig_2 = (\hat{\mathbf{I}}_2, \hat{\mathbf{z}}_{2,1}, \ldots, \hat{\mathbf{z}}_{2,l_2})$. If $\hat{\mathbf{I}}_1 = \hat{\mathbf{I}}_2$, output "link" ; otherwise, output "unlink" .

3.2 Correctness

We prove the correctness of the above scheme from two aspects: signature correctness and link correctness.

Signature Correctness: From Corollary 6.2 in [**?**], for any $\hat{\mathbf{z}} \in D$, the probability that $\|\hat{\mathbf{z}}\|_\infty \leq \beta$ is $1 - \mathrm{negl}(n)$. In our parameter setting, we have $\beta = \log n$, so the probability that the verification equation holds is overwhelming. Signature correctness can be verified by the following equation:

$$h(\hat{\mathbf{z}}_i) = h(c_i \cdot SK_{ID_i} + \hat{\mathbf{y}}_i) = c_i \cdot h(SK_{ID_i}) + h(\hat{\mathbf{y}}_i) = c_i \cdot (MPK + H(ID_i)) + \mathbf{Y}_i$$

Link Correctness: Consider two signatures Sig_1 and Sig_2 where an honest user with identity ID is the actual signer for both messages μ_1 and μ_2. If the same user is the true signer, then algorithm Link will definitely output "link" during verification. That is, if $SK_{ID}^{(1)} = SK_{ID}^{(2)}$, then $H(SK_{ID}^{(1)}) = H(SK_{ID}^{(2)})$, so $\hat{\mathbf{I}}_1 = \hat{\mathbf{I}}_2$.

In summary, the scheme satisfies correctness.

3.3 Security

We now prove the security of this lattice-based IBLRS scheme.

Lemma 1 (Anonymity). Under the random oracle model, the lattice-based ring signature scheme satisfies complete anonymity.

Proof. According to the definition of anonymity, if there exists a polynomial-time adversary $\mathcal{A}$ that can win the anonymity game in Definition 7 with non-negligible advantage ε, we can construct a challenger $\mathcal{C}$ that calls $\mathcal{A}$ as a subroutine to solve the $\mathrm{Col}(h, D)$ problem with non-negligible probability.

The interaction between $\mathcal{C}$ and $\mathcal{A}$ is as follows:

Setup: Challenger $\mathcal{C}$ determines a maximum ring user set $U_{\max}$, where max denotes the maximum number of users. $\mathcal{C}$ runs the Setup algorithm to generate public parameters PP, master public key MPK, and master secret key MSK, and sends PP and MPK to $\mathcal{A}$.

Query: Adversary $\mathcal{A}$ can make private key extraction queries and signature queries to challenger $\mathcal{C}$. Assuming $\mathcal{A}$ does not make repeated queries, $\mathcal{C}$ responds as follows: - **Extract query**: When $\mathcal{A}$ submits a user identity ID to $\mathcal{C}$, $\mathcal{C}$ runs the private key extraction algorithm Extract and returns the corresponding private key SK_{ID}. - **Sign query**: When $\mathcal{A}$ submits a ring U, message μ, and user identity ID to $\mathcal{C}$, $\mathcal{C}$ calls the ring signature algorithm RingSign to sign the message and returns the signature Sig to $\mathcal{A}$.

Challenge: After completing the query phase, $\mathcal{A}$ submits a message μ^*, ring U^*, and two user identities ID_0 and ID_1 to $\mathcal{C}$. Challenger $\mathcal{C}$ randomly selects $b \in \{0, 1\}$, runs the signature algorithm RingSign using the private key corresponding to ID_b to sign message μ^* and ring U^*, and returns the signature Sig^* to adversary $\mathcal{A}$.

Guess: Adversary $\mathcal{A}$ outputs a guess b' of the random bit b.

We now show that the advantage of $\mathcal{A}$ winning this game can be ignored. We only need to prove that the ring signature computed by challenger $\mathcal{C}$ using ID_b' s private key is statistically indistinguishable from the ring signature computed using ID_{1-b}' s private key.

By Theorem 2 [**?**], if $\hat{\mathbf{s}}$ is uniformly randomly selected from D, there exists another distribution that is statistically indistinguishable from the uniform distribution. From Definition 6 and Theorem 2, we know that $\hat{\mathbf{z}}$ and $\hat{\mathbf{z}}'$ are indistinguishable. Therefore, the scheme satisfies complete anonymity.

Lemma 2 (Strong Unforgeability). If there exists a polynomial-time adversary $\mathcal{A}$ that can output a valid forgery of this scheme with non-negligible probability ε, then using $\mathcal{A}$' s capability, we can construct a challenger $\mathcal{C}$ that can obtain a solution to the $\mathrm{Col}(h, D)$ problem with probability at least $\varepsilon^2/(2t)$, where e is the natural logarithm and t is the maximum number of hash queries allowed for the adversary.

Proof. According to the definition of strong existential unforgeability, assume there exists a polynomial-time adversary $\mathcal{A}$ that can output a valid forgery of this scheme with non-negligible advantage ε. Then we can construct challenger $\mathcal{C}$ that can solve the $\mathrm{Col}(h, D)$ problem with non-negligible probability.

The interaction between $\mathcal{C}$ and $\mathcal{A}$ is as follows:

Setup: Challenger $\mathcal{C}$ determines a maximum ring user set $U_{\max}$, where max denotes the maximum number of users. $\mathcal{C}$ runs the Setup algorithm to generate public parameters PP, master public key MPK, and master secret key MSK, and sends PP and MPK to adversary $\mathcal{A}$.

Query: Adversary $\mathcal{A}$ can make hash queries, private key extraction queries, and signature queries to challenger $\mathcal{C}$. Assuming $\mathcal{A}$ does not make repeated queries, $\mathcal{C}$ responds as follows: - **Hash query**: When $\mathcal{A}$ submits a message μ and ring U to $\mathcal{C}$, $\mathcal{C}$ returns the corresponding hash value. - **Extract query**: When $\mathcal{A}$ submits user identity ID to $\mathcal{C}$, $\mathcal{C}$ runs algorithm Extract and returns the corresponding private key SK_{ID}. - **Sign query**: When $\mathcal{A}$ submits a ring U, user identity ID, and message μ to $\mathcal{C}$, $\mathcal{C}$ calls the ring signature algorithm RingSign to sign the message and returns the signature Sig to $\mathcal{A}$.

Forge: Adversary $\mathcal{A}$ completes the above queries and outputs a valid forged signature (U^*, μ^*, Sig^*) with non-negligible probability, where $\mathcal{A}$ never made a signature query for (U^*, μ^*) and never queried the private key of any user in U^*.

The forking lemma in [**?**, **?**] shows that $\mathcal{A}$ can output two valid forgeries with non-negligible probability. Therefore, if adversary $\mathcal{A}$ successfully obtains a valid forgery of this scheme, challenger $\mathcal{C}$ can solve the $\mathrm{Col}(h, D)$ problem. According to Theorem 1 in Section 1.3 and Lemma 2 in Section 3.3, this scheme satisfies strong existential unforgeability.

Lemma 3 (Linkability). If the scheme is unforgeable, then it satisfies linkability.

Proof. According to the definition of linkability, assume there exists a polynomial-time adversary $\mathcal{A}$ that can win the linkability game in Definition 9 with non-negligible advantage ε.

The interaction between $\mathcal{C}$ and $\mathcal{A}$ is as follows:

Setup: Challenger $\mathcal{C}$ determines a maximum ring user set $U_{\max}$, where max denotes the maximum number of users. $\mathcal{C}$ runs the Setup algorithm to generate public parameters PP, master public key MPK, and master secret key MSK, and sends PP and MPK to adversary $\mathcal{A}$.

Query: Adversary $\mathcal{A}$ can make a series of hash queries and signature queries to challenger $\mathcal{C}$. Assuming $\mathcal{A}$ does not make repeated queries, $\mathcal{C}$ responds as follows: - **Hash query**: When $\mathcal{A}$ makes a hash query for (μ, U), $\mathcal{C}$ returns the corresponding hash value. - **Sign query**: When $\mathcal{A}$ submits a message μ, ring U, and user identity ID to $\mathcal{C}$, $\mathcal{C}$ returns the signature.

Forge: After completing the queries, adversary $\mathcal{A}$ outputs two ring signatures $(U_1^*, \mu_1^*, Sig_1^*)$ and $(U_2^*, \mu_2^*, Sig_2^*)$ with non-negligible probability, where: - $\mathcal{A}$ never made signature queries for (U_1^*, μ_1^*) or (U_2^*, μ_2^*); - The private keys of any members in U_1^* and U_2^* were not queried; - $\mathcal{A}$ possesses at most one private key; - Verify$(PP, U_i^*, \mu_i^*, Sig_i^*) = 1$ for $i \in \{1, 2\}$; - Link$(Sig_1^*, Sig_2^*) =$ "unlink".

Since this scheme is unforgeable, when adversary $\mathcal{A}$ honestly outputs signatures according to the rules, both signatures can only pass the verification algorithm if they use the same random oracle H output, i.e., $H(SK_{ID}^{(1)}) = H(SK_{ID}^{(2)})$. This implies $SK_{ID}^{(1)} = SK_{ID}^{(2)}$, meaning the signatures were produced by the same signer. Therefore, the link algorithm Link would output "link" , contradicting Definition 9. Thus, $\mathcal{A}$' s advantage in winning the game is negligible, and the scheme is linkable.

3.4 Efficiency Analysis

This section compares the efficiency of our scheme with existing schemes [**?**, **?**, **?**] in terms of time overhead and storage overhead.

Table 3: Time Cost Comparison

Scheme	MK	UK	Sig	Ver
Scheme [6]	-	$T_{MV} + T_{LHL}$	$3T_{MV}+2T_{SD}$	$4T_{MV}$
Scheme [21]	-	$T_{MV} + T_{SD}$	$2T_{MV} + T_{SD}$	$3T_{MV}$
Scheme [22]	T_{TG}	$T_{MV} + T_{SPT}$	$2T_{MV} + T_{SD}$	$4T_{MV}$
Our Scheme	T_{MV}	T_{MV}	T_{MV}	T_{MV}

Where l denotes the number of ring members, and T_{TG}, T_{SPT}, T_{SD}, T_{BD}, T_{LHL}, and T_{MV} represent the average time consumption of algorithms TrapGen, SamplePre, SampleDom, BasisDel, Leftover Hash Lemma (LHL), and matrix-vector operations, respectively.

Table 4: Storage Overhead Comparison

Scheme	Public Key	Private Key	Signature
Scheme [6]	$n \log q$	$m \times n \log q$	$l \cdot m \log q + n \log q$
Scheme [21]	$n \log q$	$m \log q$	$l \cdot m \log q + n \log q$
Scheme [22]	$n \log q$	$m \log q$	$(l + 1) \cdot m \log q$
Our Scheme	$n \log q$	$m \log q$	$l \cdot m \log q$

In the MK phase, both our scheme and scheme [22] involve identity-based cryptography. Scheme [22] uses the trapdoor generation algorithm to generate the

master key with time overhead T_{TG}. Our scheme does not involve trapdoor generation, with time overhead T_{MV}. Schemes [6,21] are not identity-based, so they have no such time overhead.

In the UK phase, our scheme uses a fast hash function to output the public key, followed by simple matrix-vector operations to output the private key. Therefore, public key generation time can be ignored, while private key generation time is T_{MV}. For schemes [6,21], user public keys are generated through scalar multiplication of randomly selected matrices and vectors. Scheme [6] generates private keys using LHL, while scheme [21] uses the SampleDom algorithm. Scheme [22] uses a fast hash function to generate user public keys and calls the SamplePre algorithm to generate private keys. Therefore, user key generation in these schemes requires $T_{MV} + T_{SPT}$.

In the Sig phase, our scheme generates signatures $(\hat{\mathbf{I}}, \hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_l)$, requiring only matrix-vector multiplication operations. Comparison shows that our scheme' s signature generation time overhead is much smaller than the other three schemes. In the Ver phase, only matrix-vector multiplication operations are needed. Comparison demonstrates that this scheme's signature verification efficiency is higher than the three reference schemes.

We set parameters $n = 8$, $m = 640$, $q = 2^{32}$, $p = 4294967296$, $k = 6$. The hardware environment is Windows 10 OS, AMD Ryzen 5 4600U with Radeon Graphics 2.10 GHz processor. The compilation environment is Python 3.9, JetBrains PyCharm 2018.1.3 x64. Under these conditions, simulation experiments were conducted.

Tables 5 and 6 show the time overhead and storage overhead comparison results for the reference schemes and our scheme with ring member numbers of 8, 32, and 128. Since public and private key sizes are not affected by ring member count, the storage overhead comparison focuses on signature size. Figure 1 shows the experimental results, where (a), (b), and (c) represent results for ring member counts of 8, 32, and 128, respectively. Comprehensive analysis shows that our scheme improves both time overhead and storage overhead compared to the other three reference schemes.

Table 5: Time Cost Comparison (ms)

Scheme	$l = 8$	$l = 32$	$l = 128$
	UK	Ver	UK
Scheme [6]	12.3	45.6	48.1
Scheme [21]	8.7	32.4	34.8
Scheme [22]	15.2	45.6	60.8
Our Scheme	2.1	8.4	8.4

Table 6: Signature Size Comparison (KB)

Scheme	$l = 1$	$l = 8$	$l = 32$	$l = 128$
Scheme [6]	2.5	20.0	80.0	320.0
Scheme [21]	2.5	20.0	80.0	320.0
Scheme [22]	5.0	40.0	160.0	640.0
Our Scheme	2.5	20.0	80.0	256.0

4 Conclusion

Research on lattice-based ring signatures has tremendous potential and broad prospects. However, most existing lattice-based ring signature schemes suffer from defects such as low computational efficiency and large storage overhead. Meanwhile, identity-based linkable ring signatures combine identity-based cryptography with ring signature technology, effectively reducing system overhead waste. To address potential risks from quantum algorithm attacks, our scheme combines the SVP hard problem from lattice cryptography, whose solution difficulty is equivalent to solving the collision problem on cyclic lattices. The construction process does not use sampling algorithms or trapdoor algorithms, relying only on simple matrix-vector multiplication operations, which greatly reduces computational complexity, decreases runtime at each step, and lowers storage overhead. Under the random oracle model, we provide rigorous security proofs demonstrating that the scheme satisfies anonymity, unforgeability, and linkability. Compared with existing schemes, our scheme achieves efficiency improvements in all aspects.

References

[1] Diffie W, Hellman M. New directions in cryptography [J]. IEEE Transactions on Information Theory, 1976, 22 (6): 644-654.

[2] Rivest R L, Shamir A, Tauman Y. How to leak a secret [C]// Advances in Cryptology ASIACRYPT 2001. Cambridge MA: Laboratory for Computer Science, Massachusetts Institute of Technology, 2001: 552-565.

[3] B Forum. GHash.IO and Double-Spending Against BetCoin Dice [EB/OL]. (2020-07-23) [2022-04-26]. https://bitcointalk.org/index.php?topic=327767.0.

[4] Liu J K, Wei V K, Wong D S. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups [C]// Australasian Conference on Information Security and Privacy. Berlin: Springer-Verlag, 2004: 325-335.

[5] Torres W A, Kuchta V, Steinfeld R, et al. Lattice RingCT V2.0 with Multiple Input and Multiple Output Wallets [J]. Springer, Cham, 2019, 11547: 156–175.

[6] Alberto Torres W A, Steinfeld R, Sakzad A, et al. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (Lattice RingCT v1.0) [J]. Springer, Cham, 2018, 10946: 558-578.

[7] Shen N, Mackenzie A, Lab T M. Ring confidential transactions [J]. Ledger, 2016, 1: 1–18.

[8] Sun S F, Au M H, Liu J K, et al. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero [C]// Computer Security –ESORICS 2017, LNCS. Springer, Cham. 2017, 10493: 456–474.

[9] Yuen T H, Sun S F, Liu J K, et al. RingCT 3.0 for blockchain confidential transaction: Shorter size and stronger security [C]// Financial Cryptography and Data Security. FC 2020, LNCS. Springer, Cham. 2020, 12059: 464-483.

[10] Chow S, Susilo W, Yuen T H. Escrowed linkability of ring signatures and its applications [C]// Progress in Cryptology-VIETCRYPT 2006, LNCS. Berlin: Springer. 2006, 4341: 175–192.

[11] Jeong I R, Kwon J O, Dong H L. Analysis of Revocable-iff-Linked Ring Signature Scheme [J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2009, 92 (1): 322-331.

[12] Ajtai M. Generating hard instances of lattice problems (extended abstract) [C]// Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing (STOC' 96), ACM, New York, NY, USA. 1996: 99-108.

[13] Regev O. Lattice-Based Cryptography [C]// Advances in Cryptology-CRYPTO 2006, LNCS. Berlin: Springer. 2006, 4117: 131-141.

[14] Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions [C]// Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada. 2008: 197–206.

[15] Wang Fenghe, Hu Yupu, Wang Chunxiao. A Lattice-based Ring Signature Scheme from Bonsai Trees [J]. Journal of Electronics and Information Technology. 2010, 32 (10): 2400–2403.

[16] Wang Jin, Sun Bo. Ring Signature Schemes from Lattice Basis Delegation [C]// Information and Communications Security-13th International Conference, ICICS 2011, LNCS. Berlin: Springer. 2011, 7043: 15-28.

[17] Zhang Lili, Ma Yanqin. A Lattice-Based Identity-Based Proxy Blind Signature Scheme in the Standard Model [J]. Mathematical Problems in Engineering, 2014 (1): Article ID 307637.

[18] Lai R, Cheung H, Chow S. Trapdoors for Ideal Lattices with Applications [C]// Information Security and Cryptology. Inscrypt 2014, LNCS. Springer, Cham. 2015, 8957: 239-256.

[19] Lyubashevsky V. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures [C]// International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer. 2009, 5912: 598-616.

[20] Lyubashevsky V. Lattice signatures without trapdoors [C]// Advances in Cryptology -EUROCRYPT 2012, LNCS. Berlin: Springer. 2012, 7237: 738-755.

[21] Baum C, Lin H, Oechsner S. Towards Practical Lattice-Based One-Time Linkable Ring Signatures [C]// Information and Communications Security. ICICS 2018. LNCS. Springer, Cham. 2018, 11149: 303-322.

[22] 汤永利, 夏菲菲, 叶青, 等. 格上基于身份的可链接环签名 [J]. 密码学报, 2021, 8 (2): 232-247. (Tang Yongli, Xia Feifei, Ye Qing, et al. Identity-based linkable ring signature on lattice [J]. Journal of Cryptologic Research, 2021, 8 (2): 232-247.)

[23] Lyubashevsky V, Micciancio D. Generalized Compact Knapsacks Are Collision Resistant [C]// Automata, Languages and Programming. ICALP 2006, LNCS. Berlin: Springer. 2006, 4052: 144-155.

[24] Micciancio D. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions [J]. Comput Compl, 2007, 16 (4): 365-411.

[25] Lyubashevsky V. Towards practical lattice-based cryptography [D]. University of California at San Diego, 2008.

[26] Pointcheval D, Stern J. Security Arguments for Digital Signatures and Blind Signatures [J]. Journal of Cryptology, 2000, 13 (3): 361-396.

[27] Bellare M, Neven G. Multi-signatures in the Plain public-Key Model and a General Forking Lemma [C]// Proceedings of the 13th ACM Conference on Computer and Communications Security, ACM, New York, NY, USA. 2006: 390-399.

Note: Figure translations are in progress. See original paper for figures.