cryptography

MDPI

Article

Post-Quantum Secure ID-Based (Threshold) Linkable Dual-Ring Signature and Its Application in Blockchain Transactions

Wen Gao [1,*,†], Haoyuan Yao [1,†], Baodong Qin [1], Xiaoli Dong [1], Zhen Zhao [2] and Jiayu Zeng [1]

1 School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; 15029833100@163.com (H.Y.); qinbaodong@xupt.edu.cn (B.Q.); dxl_xaut@163.com (X.D.); 17730756790@163.com (J.Z.)
2 School of Cyber Engineering, Xi'dian University, Xi'an 710071, China; zzhen@xidian.edu.cn
* Correspondence: gaowen@xupt.edu.cn
† These authors contributed equally to this work.

Abstract: Ring signatures are widely used in e-voting, anonymous whistle-blowing systems, and blockchain transactions. However, due to the anonymity of ring signatures, a signer can sign the same message multiple times, potentially leading to repeated voting or double spending in blockchain transactions. To address these issues in blockchain transactions, this work constructs an identity-based linkable ring signature scheme based on the hardness of the lattice-based Module Small Integer Solution (M-SIS) assumption, which is hard even for quantum attackers. The proposed scheme is proven to be anonymous, unforgeable, linkable, and nonslanderable in the random oracle model. Compared to existing identity-based linkable ring signature (IBLRS) schemes of linear size, our signature size is relatively smaller, and this advantage is more pronounced when the number of ring members is small. We provide approximate signature size data for ring members ranging from 2 to 2048. When the number of ring members is 16 (or 512. resp.), the signature size of our scheme is 11.40 KB (or 24.68 KB, respectively). Finally, a threshold extension is given as an additional scheme with specifications and security analysis.

Keywords: ring signature; linkability; identity-based; lattice

Citation: Gao, W.; Yao, H.; Qin, B.; Dong, X.; Zhao, Z.; Zeng, J. Post-Quantum Secure ID-Based (Threshold) Linkable Dual-Ring Signature and Its Application in Blockchain Transactions. *Cryptography* **2024**, *8*, 48. https://doi.org/10.3390/cryptography8040048

Academic Editor: Luis Javier Garcia Villalba

Received: 14 September 2024
Revised: 22 October 2024
Accepted: 23 October 2024
Published: 28 October 2024

1. Introduction

A ring signature, first proposed [1] by Rivest et al. in 2001, allows the signer to create signatures in the name of a group include him or herself (called a ring). A ring signature is verified to come from a ring, without knowing the identity of the real signer, thus ensuring the anonymity. To meet the privacy and security needs of both parties in blockchain transactions, ring signatures have been introduced to ensuring the anonymity of transaction user identities and transaction security in the last decade or so [2,3]. The first use of ring signatures on blockchains was in the Cryptonote protocol research conducted by Saberhagen et al. [4] in 2013. The Cryptonote protocol proposed two major privacy-related properties that an anonymous e-cash system needs to satisfy: untraceability and unlinkability. To meet these requirements, the protocol uses one-time public–private key pairs to protect the privacy of the recipient in transactions and, at the same time, uses one-time ring signatures to protect the privacy of the sender. This provides an important practical case and theoretical basis for the application of ring signatures in blockchain privacy protection.

However, using ring signatures to solve the privacy protection problem on blockchains also introduces the "double-spending" problem due to its anonymity. "Double spending", also known as double payment, refers to the situation where the same digital asset is used repeatedly. The linkable ring signature first introduced by Liu et al. [5] in 2004 provides a solution. By linking two legitimate ring signatures created by the same sender for a single message, the "double-spending" problem in blockchain technology can be solved.

Early linkable ring signatures were built based on Public Key Infrastructure (PKI), where certificate management issues increased computational costs. This issue can be solved by identity-based cryptography, which was proposed by Adi Shamir [6] in 1984. An identity-based signature (IBS) allows users to directly generate public keys from their identities, such as email addresses or usernames, without the need for certificates. Combining identity-based cryptography with linkable ring signature technology to achieve an identity-based linkable ring signature (IBLRS) is a significant topic that addresses identity authentication and key management issues while providing linkability to prevent repeated signatures. In 2006, Chow et al. [7] proposed the first IBLRS scheme based on the bilinear pairing assumption. Subsequently, numerous IBLRS schemes emerged [8–13].

However, many of the above schemes rely on traditional number-theoretic assumptions, including factoring and discrete logarithms, which become vulnerable to quantum attacks with the development of large-scale quantum computers [14,15]. So, researchers start to turn their attention to post-quantum cryptography [16]. Among the post-quantum candidates, lattice-based cryptography is the most promising one. This can be confirmed by the post-quantum algorithmic standards selected by NIST (National Institute of Standards and Technology) after years of analysis and argumentation [17]. Therefore, this work chooses to construct identity linkable ring signatures that rely on a lattice-based assumption to achieve post-quantum security, thus provide identity privacy, as well as linkability to avoid double spending in blockchain transactions.

1.1. Contributions

Based on the lattice-based hard problem, we propose an identity-based linkable dual-ring signature scheme as well as its threshold extension. Our proposal has several advantages:

(1) It applies identity information directly for public key operations to remove the need for certificates and third-party certificate authorities, and fully demonstrates the flexibility of identity-based keys.
(2) By adopting a dual-ring structure, it has a very short signature size, especially when the ring scale is not very large (below 2000). The "double-spending" problem in general ring signature schemes is solved by adding linkability.
(3) The scheme proposed in this paper is based on the lattice-based M-SIS assumption and can resist quantum attacks. It is proved in the random oracle model that this scheme is correct, anonymous, unforgeable, linkable, and nonslanderable.
(4) It presents a threshold extension with detailed explanations and security analysis.

1.2. Related Works

The first post-quantum one-time linkable ring signature was proposed by Torres et al. [18] in 2018. Le et al. [19] suggested IBLRS methods that rely on lattice-based SIS and ring-SIS. Tang et al. [20] proposed a new lattice-based IBLRS scheme in 2020, which reduced the signature size and computational complexity, making it more suitable for practical applications. Although lattice-based ring signature schemes offer more security and are particularly suitable for future quantum computing environments, they tend to have large signature sizes and high computational complexity. To further reduce the size of ring signatures, much effort has been made in recent years. Most schemes for reducing ring signature size use two methods: accumulators [21] and zero-knowledge proofs [22]. However, accumulators require a trusted setup. In 2021, Yuen et al. [23] introduced a novel type of ring signature known as the "dual-ring signature". This signature scheme builds upon the type-T standard signature algorithm and includes a lattice-based variant of the dual-ring signature. This new structure of ring signatures significantly reduces the signature size and speeds up the signing and verification processes compared to ordinary ring signatures. In 2024, Feng et al. [24] proposed a dual-ring signature scheme based on SM2, initially converting SM2 digital signatures into Type-T, and then integrating dual-ring with a variant of SM2 digital signatures.

2. Preliminaries

2.1. Notations

Table 1 lists the related symbols. When an integer N exists for each positive integer c and, for all $x > N$, $|f(x)| < \frac{1}{x^c}$, a function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible (*negl*). If all probabilistic polynomial-time (PPT) algorithms cannot solve a problem with a non-negligible probability, then the problem is considered *hard*.

Table 1. Symbol description.

Symbol	Description
λ	security parameter
q	an odd modulus
pp	public parameter
$\|\cdot\|$	take the square root of the sum of the squares of each element
$\|\cdot\|_\infty$	the largest absolute value among all vector elements
$\boldsymbol{A}$	matrices that form a lattice
x	representation constant
$\boldsymbol{s}_{ID_i}$	represents the signer's private key
$\mathbb{Z}$	integer set
$\mathcal{R}_q$	ring $\mathbb{Z}_q[X]/(X^d+1)$
W	set of all user identities $W = \{ID_1, \cdots, ID_N\}$
$\boldsymbol{T}_A$	the trapdoor of the lattice constituted by $\boldsymbol{A}$
$\boldsymbol{\tau}$	linking tag
Sig	represents the signature
H_1, H_2, H_3	collision-resistant hash functions

2.2. Lattices

Definition 1. *Let $\boldsymbol{B} = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n\}$ be n vectors in m-dimensional space, which are linearly independent. All integer linear combinations of the vectors in $\{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n\}$ constitute lattice $L(\boldsymbol{B})$; that is, $\Lambda = L(\boldsymbol{B}) = \{\sum_{i=1}^{n} x_i\boldsymbol{b}_i | x_i \in \mathbb{Z}\}$. We call $\{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n\}$ a basis of lattice $L(\boldsymbol{B})$.*

Definition 2 ($M\text{-}SIS_{q,n,m,\beta}$ assumption [25]). *Let q, n, m be integers and β be a positive real number. Given $\boldsymbol{A} \in \mathbb{R}_q^{n\times m}$, the Module Small Integer Solution (M-SIS) assumption aims to find a vector $\mathbf{z} \in \mathbb{R}_q^m$ such that $\boldsymbol{A}\mathbf{z} = 0$ and $\|\mathbf{z}\| < \beta$.*

The M-SIS (Module-SIS) hard problem is a modular version of the SIS (Short Integer Solution) hard problem, which transforms $\mathbb{Z}_q$ in the SIS problem to $\mathbb{R}_q$. Due to the increase in the modular structure, the M-SIS problem is more computationally complex, and finding short vectors is more challenging than in the SIS problem.

2.3. Important Algorithms

In 2008, Gentry et al. [26] proposed the GPV lattice screening algorithm, which is used by most lattice-based signature schemes and mainly consists of the following three parts:

TrapGen(1^n): Input the security parameter n; let $q = q(n) \geqslant 3$, $m = 5n\log q$, and $\sigma = \sqrt{m} \cdot 2^{\omega(\sqrt{\log m})}$. The algorithm TrapGen($1^n$) outputs a matrix $A \in \mathbb{R}_q^{n\times m}$ and a set of bases on $T \in \mathbb{R}_q^{m\times m}$, and satisfies $\widetilde{T} = O(n\log q)$.

SampleDom($1^n, \sigma$): Input the security parameter n and the Gaussian parameter σ. The algorithm SampleDom($1^n, \sigma$) selects a random vector $\boldsymbol{v} \in \mathbb{Z}^m$ according to the distribution D_σ^m, and with high probability satisfies $\| \boldsymbol{v} \| \leq \sigma\sqrt{m}$.

SamplePre($\boldsymbol{A}, \boldsymbol{T}, \sigma, \boldsymbol{y}$): Input the matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$; $\boldsymbol{T}$ is a trapdoor basis of the lattice $\Lambda^\perp(\boldsymbol{A})$; the parameter $\sigma \geq \| \tilde{\boldsymbol{T}} \| \omega(\sqrt{\log m})$; for any vector $\boldsymbol{y} \in \mathbb{Z}_q^n$, the algorithm SamplePre($\boldsymbol{A}, \boldsymbol{T}, \sigma, \boldsymbol{y}$) outputs a random non-zero vector $\boldsymbol{e} \in \mathbb{Z}_q^m$, where $\| \boldsymbol{e} \| \leq \sigma\sqrt{m}$ and $\boldsymbol{A}\boldsymbol{e} = \boldsymbol{y}(mod\ q)$.

2.4. Rejection Sampling Technique

In lattice-based digital signatures, the signer wants to output a vector z that is independent of the private key s, ensuring that z cannot be used to gain any information about the signer's secret. In the protocol, the signer computes $z = r + ls$, where s can be the private key or randomness used for the signer's secret, $l \leftarrow \mathcal{L}$ is a challenge polynomial, and r is a "masking" vector. To eliminate the dependence of z on s, rejection sampling can be applied [27].

As Theorem 1 shows, for any $\boldsymbol{v} \in \mathbb{Z}^m$, $\sigma = \omega(\| \boldsymbol{v} \| \sqrt{\log m})$, $Pr[\frac{(D_\sigma^m(z)}{(D_{\boldsymbol{v},\sigma}^m(z)} = O(1) : \boldsymbol{z} \leftarrow D_\sigma^m] = 1 - 2^{-\omega(\log m)}$.

Theorem 1. *Given a probability distribution $V = \{\boldsymbol{v} \in \mathbb{Z}^m : ||\boldsymbol{v}|| < t\}$, determine $\sigma = \omega(t\sqrt{\log m})$ and $h : V \to R$. The statistical distance between the input distributions of the next two algorithms is then less than $2^{-\omega(\log m)} / M$, where $M = O(1)$ is a constant:*

- *Distribution 1: Output $(\boldsymbol{z}, \boldsymbol{v})$ with probability $min(\frac{D_\sigma^m(z)}{MD_{v,\sigma}^m(z)}, 1)$; sample $\boldsymbol{v} \leftarrow h$ and $\boldsymbol{z} \leftarrow D_{\boldsymbol{v},\sigma}^m$;*
- *Distribution 2: With a probability of $\frac{1}{M}$, the sample $\boldsymbol{v} \leftarrow h$ and $\boldsymbol{z} \leftarrow D_\sigma^m$ yields $(\boldsymbol{z}, \boldsymbol{v})$.*

Distribution 1 has a minimum probability of producing an output of $\frac{1-2^{-\omega(\log m)}}{M}$.

2.5. The Forking Lemma

In 2000, Pointcheval and Stern proposed the forking lemma [28]. Suppose (G, Σ, V) is a digital signature scheme with security parameter n. A is a PPT algorithm whose input only consists of public data. Let Q be the maximum number of queries that A can make to the random oracle. If A generates a valid signature $(m, \sigma_1, h, \sigma_2)$ with probability $\varepsilon \geq 7Q/2^n$ within time T, then there exists an algorithm B that controls algorithm A and can generate two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ within expected time $T' \leq 84480TQ/\varepsilon$, where $h \neq h'$.

2.6. Dual-Ring Structure

To further shorten the ring signature size of the AOS structure [29], especially the number of responses, the dual-ring structure, an efficient approach for constructing ring signatures, is suggested by [23]. The dual-ring signature splits the AOS single-ring signature into two separate rings: the commitments ring and the challenges ring, which are connected using a hash function. A dual-ring signature consists of N challenges and one response. We further provide a high-level description of the dual-ring structure:

In Figure 1, *Com* represents the function used by the signer. $\odot$ and $\otimes$ are two commutative group operations. V is the verification function. The verification function is split into two parts, V_1 and V_2, and their relationship is $V = V_1 \odot V_2$. Z is the response function.

(1) The signatory selects a random number r_j and generates a commitment through the Com function.
(2) Randomly select $n - 1$ challenges c_i, where $i \in \{1, \cdots, j-1, j+1, \cdots, n\}$.
(3) Use the group operation $\odot$ and functions *Com* and V_2 to form a commitment ring.
(4) Calculate the commitment c.
(5) Link the commitment ring and the challenge ring through the hash function H_1.
(6) Obtain l_j through the hash value of H_1 and l_i, $(i \neq j)$ by group operation $\oslash$. Calculate the response z through the Z function.

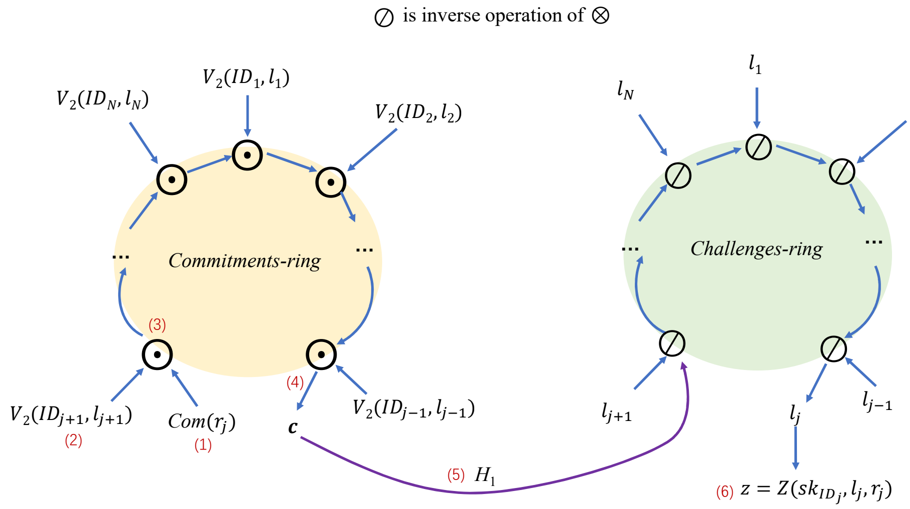

Figure 1. Structure of dual-ring structure.

3. Syntax and Security Model

As shown in the Figure 2, an identity-based linkable ring signature (IBLRS) scheme includes five PPT algorithms [20]:

(1) Setup(λ): The Key Generation Center (KGC) generates the public parameter pp and the system master private key MSK.

(2) KeyExt(ID_i, MSK, pp): Performed by the KGC, this process takes the user's identity ID_i, MSK, and pp as input, and produces the private key $\boldsymbol{s}_{ID_i}$ corresponding to the user's identity ID_i.

(3) Sign($pp, W, \boldsymbol{\mu}, \boldsymbol{s}_{ID_j}$): Operated by the signer. Taking pp, a set of ring members $W = \{ID_1, ID_2, \cdots, ID_N\}$, message $\boldsymbol{\mu}$, and the private key $\boldsymbol{s}_{ID_j}$ corresponding to the signer's identity $ID_j \in W$ as input, this algorithm outputs a linkable ring signature Sig on $\boldsymbol{\mu}$ under W. The signature Sig includes a linkable tag $\boldsymbol{\tau}$.

(4) Verify($pp, W, \boldsymbol{\mu}, Sig$): Carried out by the verifier, this process takes pp, the set of user identities W=$\{ID_1, ID_2, \cdots, ID_N\}$ forming the ring, $\boldsymbol{\mu}$, and Sig as inputs. If the verification is successful, it outputs "1"; otherwise, it outputs "0".

(5) Link($Sig, Sig', \boldsymbol{\mu}, W$): Taking as input two tuples, $(Sig, \boldsymbol{\mu}, W)$ and $(Sig', \boldsymbol{\mu}, W)$, this algorithm returns "linkable" or "unlinkable".

We illustrate the security model of IBLRS through a series of interactions between attacker $\mathcal{A}$ and challenger $\mathcal{C}$. In the context of the random oracle model (ROM), attacker $\mathcal{A}$ is granted access to the RO and can issue two distinct types of queries:

(1) Key extract query: $\mathcal{A}$ selects identity ID_i and sends it to $\mathcal{C}$ for a private key query. $\mathcal{C}$ generate $\boldsymbol{s}_{ID_i}$ corresponding to ID_i, and returns the result to $\mathcal{A}$.

(2) Signing query: $\mathcal{A}$ selects a ring signature $W = (ID_1, ID_2, \cdots, ID_N)$, a user identity $ID_j \in W, i \neq j$, and $\boldsymbol{\mu} \in \{0,1\}^*$ to send to $\mathcal{C}$ for querying. $\mathcal{C}$ returns the generated signature Sig to $\mathcal{A}$.

Definition 3 (Correctness)**.** *For any PPT attacker $\mathcal{A}$, an IBLRS scheme is correct if*

$$\Pr\left[Verify(pp, W, \boldsymbol{\mu}, Sig) = 1 \middle| \begin{array}{c} (pp, MSK) \leftarrow Setup(\lambda) \\ \boldsymbol{s}_{ID_i} \leftarrow KeyExt(ID_i, pp, MSK) \\ Sig \leftarrow Sign(pp, W, \boldsymbol{\mu}, \boldsymbol{s}_{ID_j}) \end{array} \right] = 1$$

Definition 4 (Anonymity)**.** *The anonymity of IBLRS is defined by $Game_{anony}$ below:*

(1) *System Setup: Challenger $\mathcal{C}$ inputs the security parameter λ, and the KGC generates MSK and pp. $\mathcal{C}$ sends pp to $\mathcal{A}$. $\mathcal{A}$ is allowed a polynomially bounded number of queries, each query potentially dependent on previous query results.*
(2) *Query Stage: $\mathcal{A}$ adaptively carries out various queries with polynomial time bounds.*
(3) *Challenge Phase: $\mathcal{A}$ submits the message $\boldsymbol{\mu}^*$, the ring $W = \{ID_1, ID_2, \cdots, ID_N\}$, and randomly selects the user identity $ID_b(b \in \{0,1\})$ as $\mathcal{C}$. Note that A has not queried the private key associated to ID_b. $\mathcal{C}$ returns a signature $Sig^* = (l_1^*, l_2^*, \cdots, l_N^*, \boldsymbol{z}^*, \boldsymbol{\tau}^*)$, then sends it to $\mathcal{A}$.*
(4) *Guessing Phase: $\mathcal{A}$ outputs his or her guess b'.*

The advantage of $\mathcal{A}$ in $Game_{anony}$ is defined as

$$Adv_{\mathcal{A}}^{anony} = |\Pr\{b' = b\} - 1/2|.$$

For any PPT attacker $\mathcal{A}$, an IBLRS scheme is anonymous if the advantage $Adv_{\mathcal{A}}^{anony}$ in $Game_{anony}$ is negligible.

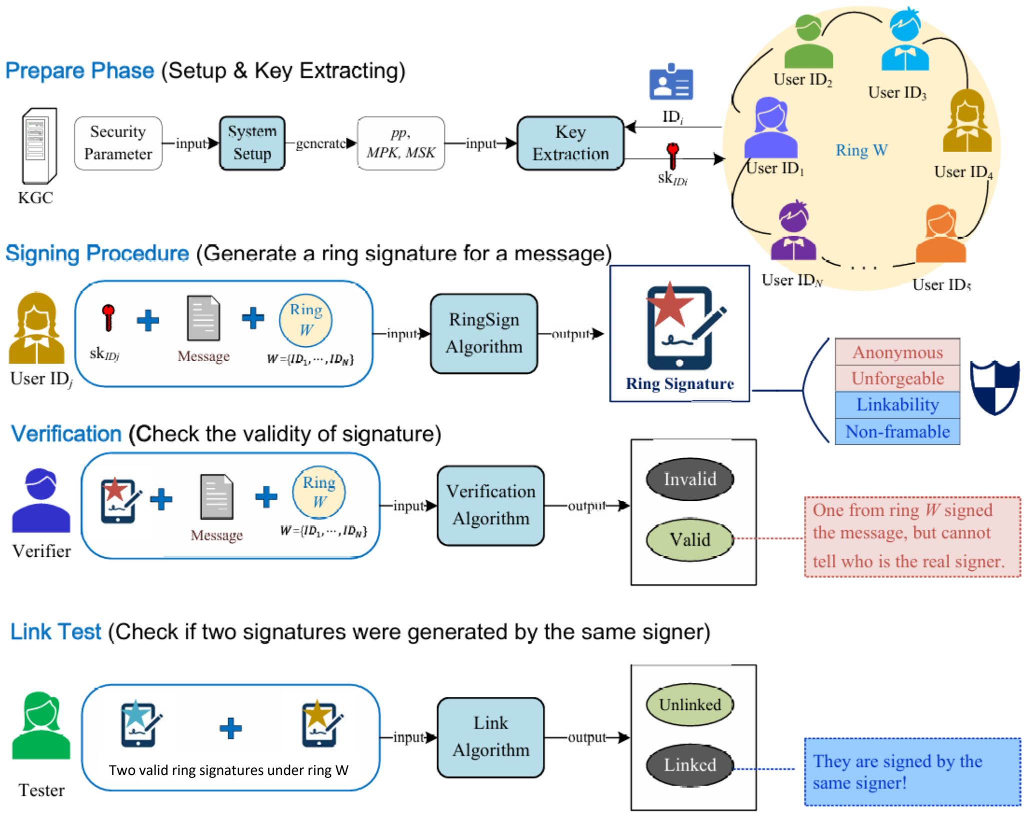

Figure 2. Definition of identity-based linkable ring signature.

Definition 5 (Unforgeability against insider corruption)**.** *We define the unforgeability of IBLRS through the $Game_{forge}$ below:*

(1) *System Setup: Challenger $\mathcal{C}$ inputs the security parameter λ, and the KGC generates MSK and pp. $\mathcal{C}$ sends pp to $\mathcal{A}$. $\mathcal{A}$ is allowed a polynomially bounded number of queries, each query potentially dependent on previous query results.*
(2) *Query Stage: $\mathcal{A}$ can access a polynomial-time oracle, and perform the aforementioned private key inquiries and signature inquiries.*
(3) *Forgery Stage: $\mathcal{A}$ provides $(\boldsymbol{\mu}^*, W^*, Sig^*)$; if it satisfies the following conditions, then the attacker $\mathcal{A}$ wins the unforgeability $Game_{forge}$:*

- $Verify(\boldsymbol{\mu}^*, W^*, Sig^*)$="1";
- *$\mathcal{A}$ has not queried the private key of any user in the ring W^*;*

- $\mathcal{A}$ *has not initiated any signature queries for* $(\boldsymbol{\mu}^*, W^*)$.

The advantage of $\mathcal{A}$ winning the unforgeability game is defined as follows:

$$Adv_{\mathcal{A}}^{forge} = \Pr[\mathcal{A}\ wins\ the\ Game_{forge}].$$

For any PPT attacker $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{forge}$ of winning $Game_{forge}$ is negligible.

Definition 6 (Linkability)**.** *We define the linkability of IBLRS through the* $Game_{link}$ *below:*

(1) *System Setup: Challenger* $\mathcal{C}$ *inputs the security parameter* λ*, and the KGC generates MSK and* pp*.* $\mathcal{C}$ *sends* pp *to* $\mathcal{A}$*.* $\mathcal{A}$ *is allowed a polynomially bounded number of queries, each query potentially dependent on previous query results.*
(2) *Query Stage:* $\mathcal{A}$ *can access a polynomial-time oracle, and perform the aforementioned private key inquiries and signature inquiries.*
(3) *Forgery Stage:* $\mathcal{A}$ *outputs two signatures* $Sig_1 = (l_1^{'}, \cdots, l_N^{'}, \boldsymbol{z}^{'}, \boldsymbol{\tau}^{'})$ *and* $Sig_2 = (l_1^{''}, \cdots, l_N^{''}, \boldsymbol{z}^{''}, \boldsymbol{\tau}^{''})$*, with linking tag* $\boldsymbol{\tau}, \boldsymbol{\tau}^{'}$*. If they satisfy the following conditions, then attacker* $\mathcal{A}$ *wins the linkability* $Game_{link}$*:*
    - $Verify(\boldsymbol{\mu}, W, Sig_i) =$ *"1",* $i \in \{1, 2\}$*;*
    - $Link(Sig_1, Sig_2) =$ *"unlinkable";*
    - *Less than two inquiries for the private key are made by attacker* $\mathcal{A}$ *(attacker* $\mathcal{A}$ *can have at most one user's private key).*

The advantage of attacker $\mathcal{A}$ winning the linkability game is defined as follows:

$$Adv_{\mathcal{A}}^{link} = Pr[\mathcal{A}\ wins\ the\ Game_{link}].$$

For any PPT attacker $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{link}$ of winning the following $Game_{link}$ is negligible.

Definition 7 (Nonslanderability)**.** *The nonslanderability of IBLRS is defined by* $Game_{NS}$ *below:*

(1) *System Setup: Challenger* $\mathcal{C}$ *inputs the security parameter* λ*, and the KGC generates MSK and* pp*.* $\mathcal{C}$ *sends* pp *to* $\mathcal{A}$*.* $\mathcal{A}$ *is allowed a polynomially bounded number of queries, each query potentially dependent on previous query results.*
(2) *Query Stage I:* $\mathcal{A}$ *can access a polynomial-time oracle, and perform the aforementioned private key inquiries and signature inquiries.*
(3) *Challenge: Attacker* $\mathcal{A}$ *sends a tuple* $(\boldsymbol{\mu}, W, \boldsymbol{\tau}, ID_b)$ *to the challenger* $\mathcal{C}$*, with the* ID_b *not having undergone a private key query. The challenger* $\mathcal{C}$ *returns a signature* Sig^**.*
(4) *Query Stage II: Similar to Query Stage I, but private key queries for* ID_b *and signature queries for* $(ID_b, \boldsymbol{\mu})$ *are not allowed.*
(5) *Slander: On* $\boldsymbol{\mu}$ *and* $\boldsymbol{\tau}$*, attacker* $\mathcal{A}$ *produces a new signature* $Sig^{'}$*. If the following scenarios are met, then attacker* $\mathcal{A}$ *wins the nonslanderability* $Game_{NS}$*:*
    - $Verify(\boldsymbol{\mu}, W, Sig^{'}) =$ *"1";*
    - $Sig^{'}$ *did not result from any queries made in Query Stage I or Query Stage II;*
    - $Link(Sig^*, Sig^{'}) =$ *"linkable".*

The advantage of $\mathcal{A}$ winning the nonslanderability game is defined as follows:

$$Adv_{\mathcal{A}}^{NS} = Pr[\mathcal{A}\ wins\ Game_{NS}].$$

For any PPT attacker $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{NS}$ of winning $Game_{NS}$ is negligible.

4. The Proposed Scheme

In this section, we first present the system model of privacy-preserving transactions on the blockchain, and then describe the construction of an identity-based linkable dual ring signature (IB-LDRS) in detail.

4.1. System Model

As Figure 3 shows, the signer with ID_i starts the transaction and creates a ring using his or her identity and the identity information of other users in the blockchain to protect anonymity in blockchain transactions. The signer signs the transaction data using their private key. Note that it is impossible for outsiders to identify which signer created the signature since the identities of the entire ring are used in the signature generation process. The ring signature and associated transaction details are broadcast along with the transaction to the blockchain network. The ring signature is validated by other nodes on the blockchain network, confirming that a member of the ring actually created it. Thus, the ring signature protects the identity privacy of its real signer.

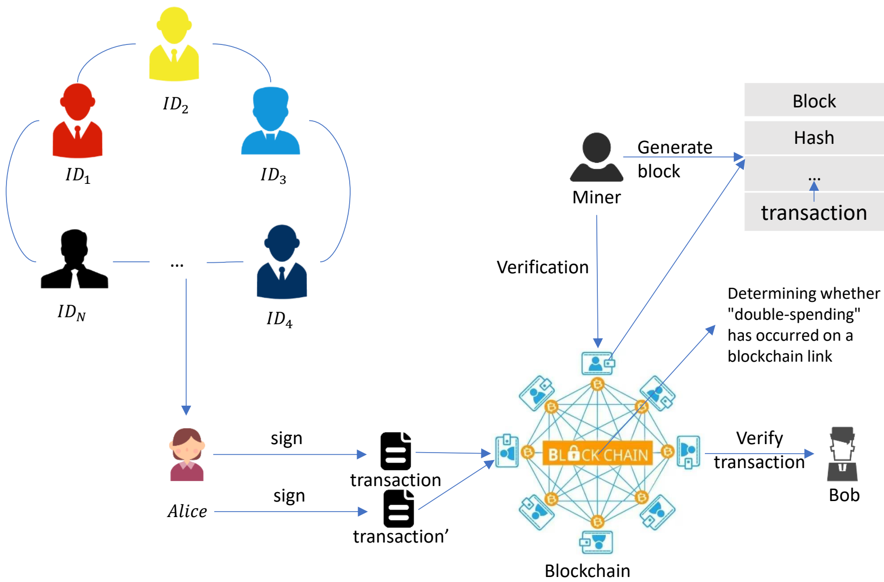

Figure 3. System model of IB-LDRS in blockchain transactions.

4.2. Parameters and Ranges

Before giving the algorithms, we first introduce the related parameters as follows: $q \geq 3$ is defined as the modulus of odd numbers, $m \geq 5n\log q$; σ_1, σ_2 are real numbers such that $\sigma_2 \geq \sigma_1$. $\mathcal{R}_q$ is a ring $\mathbb{Z}_q[X]/(X^d+1)$ of dimension d. The set D is the collection of polynomials in $\mathbb{Z}_q[X]/(X^d+1)$. We define the total ring as $W = \{ID_1, ID_2, \cdots, ID_N\}$. Define the following challenge space:

$$\mathcal{L} = \{l \in \mathbb{Z}[X]/(X^d+1) : \|l\|_\infty = 1\}$$

Observe that $|\mathcal{L}| = 3^d$. When $d = 128$, we have $|\mathcal{L}| = 3^{128} > 2^{202}$. During the computation, the polynomial coefficients need to be modulo 3. After performing the modulo operation, the polynomial coefficients will be within the range $\{-1, 0, 1\}$.

4.3. Construction

The proposed construction of the IB-LDRS scheme from lattices is described as follows:

IB-LDRS.Setup: The blockchain system executes Algorithm 1; this algorithm takes the security parameter λ as input, and outputs the public parameter pp. H_1 and H_2 act as random oracles, and H_3 functions as a collision-resistant one-way function.

Algorithm 1: IB-LDRS.Setup

Input: λ.
Output: pp.
define $H_1 : \{0,1\}^* \rightarrow \mathcal{R}_q^n$, $H_2 : \{0,1\}^* \rightarrow \mathcal{L}$, and $H_3 : \{0,1\}^* \rightarrow \mathcal{R}_q^{n\times m}$;
generate $(\boldsymbol{A}, \boldsymbol{T}_A) \leftarrow TrapGen(n,m,q)$, $\boldsymbol{A} \in \mathcal{R}_q^{n\times m}$;
define $MSK := \boldsymbol{T}_A$, $\| \widetilde{\boldsymbol{T}}_A \| \leq O(\sqrt{n \log q})$;
return $pp := \{n, m, q, \boldsymbol{A}, H_1, H_2, H_3\}$.

IB-LDRS.KeyExt: The KGC runs Algorithm 2 to generate the user's public and private keys; this algorithm takes the public parameters $pp = \{n, m, q, \boldsymbol{A}, H_1, H_2, H_3\}$, identity ID_i, and master private key MSK as inputs; compute $\boldsymbol{p}_{ID_i}$ using hash function $H_1(ID_i)$ and sample $\boldsymbol{s}_{ID_i}$ using the $SamplePre(\boldsymbol{A}, \boldsymbol{T}_A, \sigma_1, \boldsymbol{p}_{ID_i})$ function. It outputs public key ID_i and private key$\boldsymbol{s}_{ID_i}$.

Algorithm 2: IB-LDRS.KeyExt

Input: $\{ID_i, pp, MSK.\}$
Output: pk_i, sk_i.
compute $\boldsymbol{p}_{ID_i} = H_1(ID_i)$;
sample $\boldsymbol{s}_{ID_i} \leftarrow SamplePre(\boldsymbol{A}, \boldsymbol{T}_A, \sigma_1, \boldsymbol{p}_{ID_i})$ with trapdoor $\boldsymbol{T}_A$, where $\sigma_1 \geq \| \widetilde{\boldsymbol{T}}_A \| \, \omega(\sqrt{\log q})$, $\boldsymbol{s}_{ID_i} \leq \sigma_1 \sqrt{md}$;
return: $(pk_i, sk_i) = (ID_i, \boldsymbol{s}_{ID_i})$.

IB-LDRS.Sign: The transaction initiator, Alice, runs the signature Algorithm 3 to initiate a transaction; the following algorithm generates the ring signature of message $\boldsymbol{\mu}$ based on the dual-ring architecture, given input $\boldsymbol{\mu}$, pp, W. The signer's index is $j, (1 \leq j \leq N), \boldsymbol{s}_{ID_j}$.

Algorithm 3: IB-LDRS.Sign

Input: $pp, W, \boldsymbol{\mu}, \boldsymbol{s}_{ID_j}$
Output: Sig.
1 compute $\boldsymbol{A}_{com} = H_3(\boldsymbol{A}, \boldsymbol{\mu})$;
2 define $\boldsymbol{\tau} := \boldsymbol{A}_{com} \cdot \boldsymbol{s}_{ID_j}$;
3 pick $\boldsymbol{r} \leftarrow D^m_{\sigma_2}, \boldsymbol{e} = \boldsymbol{A} \cdot \boldsymbol{r} \in \mathcal{R}_q^n, l_i \leftarrow \mathcal{L}, i \in \{1, \cdots, N\}, i \neq j$;
4 compute $\boldsymbol{c} = \boldsymbol{e} - \sum_{i=1, i\neq j}^{N} l_i \cdot H_1(ID_i)$
5 compute $\boldsymbol{u} = \boldsymbol{A}_{com} \cdot \boldsymbol{r} - \sum_{i=1, i\neq j}^{N} l_i \cdot \boldsymbol{\tau}$;
6 compute $l_j = H_2(\boldsymbol{c}, \boldsymbol{\mu}, W, \boldsymbol{u}) - \sum_{i=1, i\neq j}^{N} l_i$;
7 define and compute $\boldsymbol{z} := \boldsymbol{r} + l_j \cdot \boldsymbol{s}_{ID_j}$;
8 if $\|\boldsymbol{z}\| > (\sigma_1 + \sigma_2)\sqrt{md}$, then restart from step 3;
9 return $Sig = (l_1, l_2, \cdots, l_N, \boldsymbol{z}, \boldsymbol{\tau})$

IB-LDRS.Verify: The transaction receiver, Bob, runs the verification Algorithm 4 to verify the transaction; given pp, W, $\boldsymbol{\mu}$, and Sig, verify it by the following steps.

Algorithm 4: IB-LDRS.Verify

Input: $pp, W, \boldsymbol{\mu}, Sig$
Output: 0 or 1.
if $l_i \notin \mathcal{L}$, return 0
if $\|\boldsymbol{z}'\| > (\sigma_1 + \sigma_2)\sqrt{md}$, return 0
$\boldsymbol{A}_{com} = H_3(\boldsymbol{A}, \boldsymbol{\mu})$
$\boldsymbol{c}' = \boldsymbol{A} \cdot \boldsymbol{z}' - \sum_{i=1}^{N} l_i \cdot H_1(ID_i)$
$\boldsymbol{u}' = \boldsymbol{A}_{com} \cdot \boldsymbol{z}' - \sum_{i=1}^{N} l_i \cdot \boldsymbol{\tau}$
check if $\sum_{i=1}^{N} l_i = H_2(\boldsymbol{c}', \boldsymbol{\mu}, W, \boldsymbol{u}')$, if so return 1
else return 0

IB-LDRS.Link: The blockchain link runs the linking Algorithm 4 to conduct "double-spending" detection. Once it receives two different and valid signatures $Sig, Sig^{'}$ to be tested, this algorithm checks whether $\boldsymbol{\tau} \stackrel{?}{=} \boldsymbol{\tau}^{'}$. If it does, it returns "linkable"; otherwise, it returns "unlinkable". Note that this algorithm tests valid signatures only, because it can invoke Algorithm 4 to check the validity and reject the invalid signatures. If both of the signatures are accepted, go to Algorithm 5 to check whether they are generated by the same signer.

Algorithm 5: IB-LDRS.Link

Input: $Sig, Sig^{'}$
Output: linkable or unlinkable.
1 if $\boldsymbol{\tau} = \boldsymbol{\tau}^{'}$, return "linkable"
2 else, return "unlinkable"

5. Security Analysis

Theorem 2 (Correctness). *A linkable ring signature generated by a legitimate signature system can pass the verification of the algorithm, thereby satisfying the correctness verification.*

Since it is impossible to determine whether it has been tampered with during transmission, suppose $Sig^{'} = (l_1^{'}, l_2^{'}, \cdots, l_N^{'}, \boldsymbol{z}^{'}, \boldsymbol{\tau}^{'})$ is the signature received by the IB-LDRS.Verify Algorithm 4, $\boldsymbol{z} = \boldsymbol{r} + l_j \cdot \boldsymbol{s}_{ID_j}$, and it will be accepted by the IB-LDRS.Verify Algorithm 4 as follows.

Correctness of $\boldsymbol{u}$:

$$\begin{aligned}
\boldsymbol{u}^{'} &= \boldsymbol{A}_{com} \cdot \boldsymbol{z}' - \textstyle\sum_{i=1}^{N} l_i' \cdot \boldsymbol{\tau} \\
&= \boldsymbol{A}_{com} \cdot (\boldsymbol{r} + l_j' \cdot \boldsymbol{s}_{ID_j}) - \textstyle\sum_{i=1}^{N} l_i' \cdot \boldsymbol{\tau} \\
&= \boldsymbol{A}_{com} \cdot \boldsymbol{r} + l_i' \cdot \boldsymbol{A}_{com} \cdot \boldsymbol{s}_{ID_i} - \textstyle\sum_{i=1}^{N} l_i' \cdot \boldsymbol{\tau} \\
&= \boldsymbol{A}_{com} \cdot \boldsymbol{r} - \textstyle\sum_{i=1, i \neq j}^{N} l_i' \cdot \boldsymbol{\tau} \\
&= \boldsymbol{u}
\end{aligned}$$

Correctness of Sig:

$$\begin{aligned}
\boldsymbol{c}^{'} &= \boldsymbol{A} \cdot \boldsymbol{z}' - \textstyle\sum_{i=1}^{N} l_i' \cdot H_1(ID_i) \\
&= \boldsymbol{A} \cdot (\boldsymbol{r} + l_j' \cdot \boldsymbol{s}_{ID_j}) - \textstyle\sum_{i=1}^{N} l_i' \cdot H_1(ID_i) \\
&= \boldsymbol{A} \cdot \boldsymbol{r} + l_j' \cdot \boldsymbol{A} \cdot \boldsymbol{s}_{ID_j} - \textstyle\sum_{i=1}^{N} l_i' \cdot H_1(ID_i) \\
&= \boldsymbol{A} \cdot \boldsymbol{r} - \textstyle\sum_{i=1, i \neq j}^{N} l_i' \cdot H_1(ID_i) \\
&= \boldsymbol{c}
\end{aligned}$$

Therefore, $\sum_{i=1}^{N} l_i = H_2(\boldsymbol{c}^{'}, \boldsymbol{\mu}, W, \boldsymbol{u}^{'})$ holds, and the proposed scheme meets the correctness requirement.

Theorem 3 (Unforgeability). *Under the assumption of M-SIS$_{n,m+1,q,\beta}$, for any PPT attacker $\mathcal{A}$, the scheme is unforgeable under chosen message attacks and insider corruption attacks in the random oracle model, where $\beta \leq 3(\sigma_1 + \sigma_2)\sqrt{md+1}$.*

Proof. Let us assume that there is an attacker $\mathcal{A}$ with a non-negligible advantage ε that can forge signatures in polynomial time. Then, there is a challenger $\mathcal{C}$ that has a non-negligible probability of solving the M-SIS hard problem. Assume $\mathcal{C}$ has an M-SIS$_{n,m+1,q,\beta}$ instance

$(\hat{\boldsymbol{A}}, n, m, q, \beta)$ to solve, where $\hat{\boldsymbol{A}} \in \mathbb{R}_q^{n\times(m+1)}$. Finding a short vector e such that $\hat{\boldsymbol{A}}\boldsymbol{e} = \mathbf{0}$ mod q that $||\mathbf{e}|| \leq \beta$ is the aim of $\mathcal{C}$. $\mathcal{C}$ first transforms $\hat{\boldsymbol{A}}$ into the form $[\mathbf{A}||\boldsymbol{a}]$, and then embeds it in the reduction algorithm. The hash functions H_1 and H_2 are random oracles. $\mathcal{C}$ establishes four lists, L_1, L_2, L_3, and L_4, which are used to store H_1-oracle, H_2-oracle, signature queries, and corruption queries, respectively. The following describes how $\mathcal{C}$ and $\mathcal{A}$ interact:

- IB-LDRS.Setup Stage: Generate system parameter pp. Send the system parameters $pp = \{n, m, q, \boldsymbol{A}, H_1, H_2, H_3\}$ and the ring W to $\mathcal{A}$.
- Query phase: During this phase, the attacker $\mathcal{A}$ interacts with $\mathcal{C}$ by making oracle queries to learn information about the scheme. The challenge $\mathcal{C}$ responds to the queries as follows.

  (1) H_1 oracle query: When $\mathcal{A}$ submits user $ID_i (i \in [N])$ to $\mathcal{C}$, for $i \neq j^*$, $\mathcal{C}$ checks whether $(ID_i, *, *)$ exists in L_1: if so, it returns $\boldsymbol{s}_{ID_i}$ to $\mathcal{A}$; if not, $\mathcal{C}$ randomly selects $\boldsymbol{s}_{ID_i} \in \mathcal{D}_\sigma^m$, and then computes $\boldsymbol{p}_{ID_i} = \boldsymbol{A}\cdot\boldsymbol{s}_{ID_i}$, assigns $\boldsymbol{p}_{ID_i}$ to $H_1(ID_i)$, and returns it to $\mathcal{A}$. $\mathcal{C}$ records it in list $L_1 = (ID_i, \boldsymbol{s}_{ID_i}, H_1(ID_i))$. If $i = j^*$, $\mathcal{C}$ sets

  $$H_1(ID_{j^*}) = \boldsymbol{A}\cdot\boldsymbol{s} + \boldsymbol{a} \tag{1}$$

  for randomly chosen $\boldsymbol{s} \in \mathcal{D}_\sigma^m$, and returns $H(ID_{j^*})$ to $\mathcal{A}$; $\mathcal{C}$ records it in list $L_1 = (ID_{j^*}, \boldsymbol{s}, H(ID_{j^*}))$.

  (2) H_2 oracle query: Upon $\mathcal{C}$ receiving an H_2 oracle query with message $\boldsymbol{\mu}$, ring $W' \in \{ID_1, ID_2, \cdots, ID_N\}$, and intermediate parameters R and T to $\mathcal{C}$ from $\mathcal{A}$, $\mathcal{A}$ first searches $(\boldsymbol{c}, \boldsymbol{\mu}, W, \boldsymbol{u}, *)$ in list L_2; if found, it returns the corresponding hash value to $\mathcal{A}$; if not, it randomly choose a vector $l \leftarrow \mathcal{L}$, and returns l to $\mathcal{A}$. Finally, $\mathcal{C}$ records $(\boldsymbol{c}, \boldsymbol{\mu}, W, \boldsymbol{u}, l)$ in list L_2. This query can be made at most q_H times.

  (3) Registration query: When $\mathcal{A}$ sends a new identity $ID_i \notin W$ for registration, $\mathcal{C}$ first randomly chooses $\boldsymbol{s}_{ID_i}$ and computes $H_1(ID_i) = \boldsymbol{A}\cdot\boldsymbol{s}_{ID_i}$ as for the H_1 oracle, and then returns the private key $\boldsymbol{s}_{ID_i}$ to $\mathcal{A}$. Finally, $\mathcal{C}$ adds ID_i to list L_4 and tuple $(ID_i, \boldsymbol{s}_{ID_i}, H_1(ID_i))$ to list L_1.

  (4) Signing oracle query: When $\mathcal{A}$ submits an inquiry for a ring signature on identity ID_j of message $\boldsymbol{\mu}$ under ring W' such that $ID_j \in W'$, if $j = j^*$, $\mathcal{C}$ chooses random $\mathbf{z}$ with $||\mathbf{z}|| \leq (\sigma_1 + \sigma_2)\sqrt{md}$, and random $l_1, \cdots, l_{j-1}, l_{j+1}, \cdots, l_N \in \mathcal{L}$, and computes $\boldsymbol{c'}$ and $\boldsymbol{u'}$ as in the verification algorithm. $\mathcal{C}$ calculates l_j through $l_j = H_2(\boldsymbol{c'}, \boldsymbol{\mu}, W, \boldsymbol{u'}) - \sum_{i=1, i\neq j}^N l_i$, and stores it in L_2, and then returns the signature $Sig = (l_1, \cdots, l_N, \boldsymbol{z}, \boldsymbol{\tau})$, where $\boldsymbol{\tau} = H_3(\boldsymbol{A})\boldsymbol{s}$. If $j \neq j^*$, $\mathcal{C}$ first checks if $W' \in \{ID_1, ID_2, \cdots, ID_N\} \cup L_4$. If not, it returns $\perp$ to $\mathcal{A}$. If it does meet the condition, $\mathcal{C}$ directly checks tuple $(\boldsymbol{\mu}, ID_j, W', *)$ in list L_3 and returns the signature to $\mathcal{A}$ if it does exist. Otherwise, $\mathcal{C}$ generates a new signature as in the following steps. If it does exist, $\mathcal{C}$ researches $(ID_j, *, *)$ in list L_1. If it exists, $\mathcal{C}$ generates a ring signature $Sig = (l_1, \cdots, l_N, \boldsymbol{z}, \boldsymbol{\tau})$ of $\boldsymbol{\mu}$ under W' with $\boldsymbol{s}_{ID_i}$ by the steps in the signing algorithm. If $(ID_i, *, *)$ does not exist in list L_1, $\mathcal{C}$ invokes the H_1 oracle to achieve the private key and then generates ring signature Sig as before. Note that tuple $(\boldsymbol{c}, \boldsymbol{\mu}, W', \boldsymbol{u}, l)$ should have been added to list L_2 by the query to H_2 during the generation of the ring signature, where $\boldsymbol{c}$ is an intermediate value in signing procedures. Finally, $\mathcal{C}$ returns the signature Sig of message $\boldsymbol{\mu}$ under ring W', and then stores tuple $(\boldsymbol{\mu}, ID_j, W', Sig)$ in list L_3.

  (5) Corruption query: If $\mathcal{A}$ selects a user identity $ID_i (i \in [N], i \neq j)$ to corrupt, $\mathcal{C}$ first checks whether ID_i exists in list L_4. If it does, $\mathcal{C}$ searches $(ID_i, *, *)$ in list L_1 and returns the corresponding private key $\boldsymbol{s}_{ID_i}$ to $\mathcal{A}$; if it does not, $\mathcal{C}$ randomly choose $\boldsymbol{s}_{ID_i}$ and generates $H_1(ID_i)$ as for the H_1 oracle, and then returns the private key $\boldsymbol{s}_{ID_i}$ to $\mathcal{A}$. Finally, $\mathcal{C}$ adds ID_i to list L_4, and tuple

$(ID_i, \boldsymbol{s}_{ID_i}, H_1(ID_i))$ to list L_1. If $\mathcal{A}$ selects a user identity $ID_i(i \in [N], i \neq j)$ to corrupt, $\mathcal{C}$ fails and aborts.

- Forgery Stage: After polynomial queries to the oracles, $\mathcal{A}$ submits a signature $Sig^* = (l_1^*, l_2^*, \cdots, l_N^*, \boldsymbol{z}^*, \boldsymbol{\tau}^*)$ of message $\boldsymbol{\mu}$ under ring W^* as his or her forgery to challenger $\mathcal{C}$. The signature Sig^* is considered to be a successful forgery if it satisfies the following conditions:
  (1) Attacker $\mathcal{A}$ never registers or corrupts any user $ID_i^* \in W^*$, that is, $W^* \cap L_4 = \emptyset$;
  (2) Attacker $\mathcal{A}$ has not queried the signature of $\boldsymbol{\mu}^*$ under W^*, that is, $(\boldsymbol{\mu}^*, W^*, Sig^*) \notin L_3$;
  (3) The forgery $(\boldsymbol{\mu}^*, W^*, Sig^*)$ can pass the verification algorithm, that is, IB-LDRS.Verify$(\boldsymbol{\mu}^*, W^*, Sig^*)$="1".

Analysis: Assume $\sigma^* = (l_1^*, l_2^*, \cdots, l_N^*, \boldsymbol{z}^*, \boldsymbol{\tau}^*)$ is a successful forgery with probability ε; then, the verification equation

$$\boldsymbol{c}^* = \boldsymbol{A}\boldsymbol{z}^* - \sum_{ID_i \in W^*} l_i^* \cdot H_1(ID_i) \tag{2}$$

holds from the correctness property. There must be one $l_i^* \in \{l_1^*, l_2^*, \cdots, l_N^*\}$ that comes from the response of oracle H_2, so $(\boldsymbol{c}^*, \boldsymbol{\mu}^*, W^*, \boldsymbol{u}^*, l_j^*)$ can be found in list L_2. From the general forking lemma [28], $\mathcal{C}$ can obtain another valid signature $Sig' = (l_1', l_2', \cdots, l_N', \boldsymbol{z}', \boldsymbol{\tau}')$ where $Sig' \neq Sig^*$ with same randomness from $\mathcal{A}$ of message $\boldsymbol{\mu}^*$ under W^* by rewinding the random oracle H_2, with a probability at least $\frac{\varepsilon}{q_H} - \frac{1}{3^d}$. So, $l_i' = l_i^*$ for $i \neq j$, $\boldsymbol{c}' = \boldsymbol{c}^*$, $\boldsymbol{r}^* = \boldsymbol{r}'$,$l_j' \neq l_j^*$. The verification equation

$$\boldsymbol{c}' = \boldsymbol{A}\boldsymbol{z}' - \sum_{ID_i \in W'} l_i' \cdot H_1(ID_i) \tag{3}$$

holds by the correctness property. Subtracting Equation (2) from Equation (3) yields the following equation:

$$(l_{j'}' - l_j^*)H_1(ID_j) = \boldsymbol{A}(\boldsymbol{z}^* - \boldsymbol{z}') + (l_{j'}' - l_j^*)\boldsymbol{A}\boldsymbol{r}^* \tag{4}$$

$$= \hat{\boldsymbol{A}}\left[\begin{pmatrix} \boldsymbol{z}^* - \boldsymbol{z}' \\ 0 \end{pmatrix} + (l_{j'}' - l_j^*)\begin{pmatrix} \boldsymbol{r}^* \\ 0 \end{pmatrix}\right] \tag{5}$$

Then, we multiply Equation (1) by $(l_{j'}' - l_j^*)$ to achieve

$$(l_{j'}' - l_j^*)H_1(ID_j) = \boldsymbol{A}(l_{j'}' - l_j^*)\boldsymbol{s} + (l_{j'}' - l_j^*)\boldsymbol{a} \tag{6}$$

$$= \hat{\boldsymbol{A}}(l_{j'}' - l_j^*)\begin{pmatrix} \boldsymbol{s} \\ 1 \end{pmatrix} \tag{7}$$

By subtracting Equation (7) from Equation (5), we obtain a short $\boldsymbol{e}$ such that

$$\boldsymbol{e} = (l_{j'}' - l_j^*)\begin{pmatrix} \boldsymbol{s} \\ 1 \end{pmatrix} - \left[\begin{pmatrix} \boldsymbol{z}^* - \boldsymbol{z}' \\ 0 \end{pmatrix} + (l_{j'}' - l_j^*)\begin{pmatrix} \boldsymbol{r}^* \\ 0 \end{pmatrix}\right].$$

$\boldsymbol{e}$ is a non-zero vector as its last coordinate is $(l_{j'}' - l_j^*)$ which is not zero. Therefore, $\mathcal{C}$ can output a valid solution to M-SIS$_{n,m+1,q,\beta}$, where $\beta = 3(\sigma_1 + \sigma_2)\sqrt{md+1}$. Therefore, we can conclude that our signature algorithm is strongly unforgeable under the message chosen and the insider corruption attack. This completes the proof. □

Theorem 4 (Anonymity). *The proposed IB-LDRS scheme satisfies unconditional anonymity.*

Proof. The challenger $\mathcal{C}$ and the PPT attacker $\mathcal{A}$ interact in a game to prove the scheme's anonymity. The attacker $\mathcal{A}$ provides $\mathcal{C}$ with a message, two identities, and a ring, after which $\mathcal{C}$ returns a signature. If $\mathcal{A}$ can guess the identity of the signer with a non-negligible probability, the scheme's anonymity is compromised.

- IB-LDRS.Setup Stage: Determine the ring $W^* = \{ID_1, ID_2, \cdots, ID_N\}$. Challenger $\mathcal{C}$ generates pp and W for each user. Then, $\mathcal{C}$ sends $pp = \{n, m, q, \boldsymbol{A}, H_1, H_2, H_3\}$ to $\mathcal{A}$.
- Query Stage: Conduct various queries adaptively on $\mathcal{C}$ with polynomial time limits.
- Challenge Stage: $\mathcal{A}$ submits a message $\boldsymbol{\mu}$, ring $W^* = \{ID_1^*, ID_2^*, \cdots, ID_N^*\}$, and user identity ID_b to $\mathcal{C}$. $\mathcal{C}$ randomly selects $b \in \{0, 1\}$, computes for $i \neq j$, $l_i^* \leftarrow \mathcal{L}$, $l_j^* = H_2(\boldsymbol{c}, \boldsymbol{\mu}, W^*, \boldsymbol{u}) - \sum_{i=1, i\neq j}^{N} l_i^*$, $\boldsymbol{z}^* = \boldsymbol{r}^* - \boldsymbol{s}_{ID_j}^* l_j^*$, and performs a ring signature $Sig^* = (l_1^*, \cdots, l_N^*, \boldsymbol{z}^*, \boldsymbol{\tau}^*)$, then sends it to $\mathcal{A}$.
- Guess Stage: $\mathcal{A}$ outputs the guess b'.
- Forgery Stage: To demonstrate that the probability $Adv_{\mathcal{A}}^{anon} = |Pr[b' = b] - 1/2| = \varepsilon$ of $\mathcal{A}$ winning the game is negligible, we only need to prove that the signature $Sig^* = \{l_1^*, \cdots, l_N^*, \boldsymbol{z}^*, \boldsymbol{\tau}^*\}$ generated by ID_b and the signature $Sig' = \{l_1', \cdots, l_N', \boldsymbol{z}', \boldsymbol{\tau}'\}$ generated by ID_{1-b} are statistically indistinguishable.

When signing Sig^*, $i \neq b$ results in $l_i \leftarrow \mathcal{L}$, and $i = b$ results in $l_b = H_2(\boldsymbol{c}, \boldsymbol{\mu}, W, \boldsymbol{u}) - \sum_{i=1, i\neq j}^{N} l_i \cdot \tau, \boldsymbol{z}^* = r^* + \boldsymbol{s}_{ID_b}$, according to Theorem 1, $\boldsymbol{z}^*$ and the Gaussian distribution D_σ^{m+1} are statistically indistinguishable; thus, the signature Sig^* is statistically indistinguishable from D_σ^{m+1}. Similarly, the signature Sig' is also statistically indistinguishable from D_σ^{m+1}. Therefore, Sig^* and Sig' follow the same discrete Gaussian distribution, making them statistically indistinguishable. Consequently, the probability that $\mathcal{A}$ can determine whether Sig^* was generated by ID_0 or ID_1 is negligible. □

Theorem 5 (Linkability)**.** *For any polynomial-time attacker $\mathcal{A}$, the proposed IB-LDRS scheme is linkable in the ROM.*

Proof. The linkability of the scheme is proved by an interactive security game between challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$.

- IB-LDRS.Setup Stage: Challenger $\mathcal{C}$ inputs the security parameter λ. Generate the public parameter pp. Send the system parameter pp to the attacker $\mathcal{A}$.
- Inquiry Stage: Same as in the scheme's unforgeability proof.
- Challenge Stage I: The attacker $\mathcal{A}$ provides two signatures, denoted $Sig_1 = (l_1', \cdots, l_N', \boldsymbol{z}', \boldsymbol{\tau}')$ and $Sig_2 = (l_1'', \cdots, l_N'', \boldsymbol{z}'', \boldsymbol{\tau}'')$.

Analysis. Attacker $\mathcal{A}$ uses a single private key to generate two ring signatures Sig_1 and Sig_2 for the same message with a non-negligible probability. These signatures can pass the verification algorithm and satisfy $\boldsymbol{\tau}' \neq \boldsymbol{\tau}''$.

$$\begin{cases} \boldsymbol{c}' = \boldsymbol{A}\boldsymbol{z}' - \sum_{i=1}^{N} l_i' \cdot H_1(ID_i) & (8) \\ \boldsymbol{u}' = \boldsymbol{A}_{com}\boldsymbol{z}' - \sum_{i=1}^{N} l_i' \cdot \boldsymbol{\tau}' & (9) \end{cases}$$

$$\begin{cases} \boldsymbol{c}'' = \boldsymbol{A}\boldsymbol{z}'' - \sum_{i=1}^{N} l_i'' \cdot H_1(ID_i) & (10) \\ \boldsymbol{u}'' = \boldsymbol{A}_{com}\boldsymbol{z}'' - \sum_{i=1}^{N} l_i'' \cdot \boldsymbol{\tau}'' & (11) \end{cases}$$

We assume that $\boldsymbol{c}'$, $\boldsymbol{c}''$, and $\boldsymbol{u}'$ are generated by the signer's own private key, while $\boldsymbol{u}''$ is forged. Simplifying the operations Equations (10) and (11) yields the following:

$$\begin{cases} \boldsymbol{A}\boldsymbol{r} - \sum_{i=1, i \neq j}^{N} l_i'' \cdot H_1(ID_i) = \boldsymbol{A}\boldsymbol{z}'' - \sum_{i=1}^{N} l_i'' \cdot H_1(ID_i) & (12) \\ \boldsymbol{A}_{com}\boldsymbol{r} - \sum_{i=1, i \neq j}^{N} l_i'' \cdot \boldsymbol{\tau}'' = \boldsymbol{A}_{com}\boldsymbol{z}'' - \sum_{i=1}^{N} l_i'' \cdot \boldsymbol{\tau}'' & (13) \end{cases}$$

$$\begin{cases} \boldsymbol{A} \cdot l_j'' \cdot (\boldsymbol{s}_{ID_j} - \boldsymbol{s}_{ID_j}) = \boldsymbol{0} & (14) \\ \boldsymbol{A}_{com} \cdot l_j'' \cdot (\boldsymbol{s}_{ID_j} - \boldsymbol{s}'_{ID}) = \boldsymbol{0} & (15) \end{cases}$$

From the above equations, through simple derivation, we can obtain $\boldsymbol{s}_{ID_j} = \boldsymbol{s}'_{ID}$, which leads to $\boldsymbol{\tau}' = \boldsymbol{\tau}''$. This contradicts the assumption; thus, the signatures of the same signer on the same message can be linked.

In the event that the signer did not utilize their private key in Sig_2, then the signature is legitimately faked. According to the unforgeability of Theorem 3, challenger $\mathcal{C}$ can generate Sig_2^* using the forking lemma based on forger $\mathcal{A}$'s ability. Subtracting $\boldsymbol{c}''$ from $\boldsymbol{c}^*$ yields $\boldsymbol{A}(\boldsymbol{z}'' - \boldsymbol{z}^*) = \boldsymbol{0}$; thus, we obtain a solution to the M-SIS hard problem. Therefore, we can conclude that legitimate signatures generated for the same message by the same signer are linkable. This completes the proof. □

Theorem 6 (Nonslanderability)**.** *The IB-LDRS is nonslanderable in the random oracle model, if the M-SIS problem is hard.*

Proof. Challenger $\mathcal{C}$ and polynomial-time attacker $\mathcal{A}$ interact in a game to prove the nonslanderability of the scheme. We will explain that the nonslanderability relies on the scheme's unforgeability.

In the security model of nonslanderability, attacker $\mathcal{A}$ sends a tuple $(\boldsymbol{\mu}, W, \boldsymbol{\tau}, ID_b)$ to challenger $\mathcal{C}$, with the ID_b not having undergone a private key query. Challenger $\mathcal{C}$ obtains the private key $\boldsymbol{s}_{ID_b}$ for the ID by running IB-LDRS.KeyExt$(ID_b, pp, MSK, \boldsymbol{\tau})$. Then, challenger $\mathcal{C}$ runs IB-LDRS.Sign$(pp, W, \boldsymbol{\mu}, \boldsymbol{s}_{ID_b})$ to obtain the signature Sig^*. On the same message $\boldsymbol{\mu}$ and tag $\boldsymbol{\tau} = \boldsymbol{\tau}^{'}$, attacker $\mathcal{A}$ produces a new signature $Sig^{'}$.

This implies that, for any PPT attacker $\mathcal{A}$, if he or she knows $\boldsymbol{s}_{ID_\pi} \in W \setminus \{ID_b\}$, he or she can produce a signature with the linkability tag τ without knowing the private key $\boldsymbol{s}_{ID_b}$. According to the unforgeability of Theorem 3, challenger $\mathcal{C}$ can generate $Sig^{''}$ using the forking lemma based on forger $\mathcal{A}$'s ability. Subtracting $\boldsymbol{c}''$ from $\boldsymbol{c}'$ yields $\boldsymbol{A}(\boldsymbol{z}'' - \boldsymbol{z}') = \boldsymbol{0}$; thus, we obtain a solution to the M-SIS hard problem. Consequently, we can state that legitimate signatures generated by the same signer for the same message ought to be connected. □

6. Performance Analysis

In this section, we will compare our scheme with other ring signature schemes, including functionality, computational overhead, and communication overhead.

6.1. Functionality Comparison

We compare the scheme's functionality with those of other schemes in the section below. Table 2 compares five features, including post-quantum resistant (PQR), linkability (Link), identity-based (ID-based), dual-ring (DR), and hard problem assumptions (Assumption). Unlike the Yuen et al. [23] lattice-based dual-ring signature system from 2019, our scheme improves linkability and resolves the blockchain's "double-spending" problem. The SM2-based dual-ring scheme proposed by Feng et al. [24] in 2024 is similar in structure to our scheme, but it does not possess quantum-resistant properties. Tang et al. presented [26], a scheme based on the NTRU lattice that satisfies the properties of

PQR, Link, and ID-based. The two schemes [30,31] only satisfy the PQR and ID-based properties. The schemes in [20,32,33] satisfy all properties except DR. Our scheme satisfies all the functionalities aforementioned.

Table 2. Comparison of functionality.

Scheme	PQR	Link	ID-Based	DR	Assumption
[23]	√	×	×	√	M-SIS
[24]	×	√	×	√	DDH
[26]	√	√	√	×	NTRU-SIS
[30]	√	×	√	×	SIS
[31]	√	×	√	×	SIS&LWE
[33]	√	√	√	×	M-LWE&M-SIS
[20]	√	√	√	×	SIS
[32]	√	√	√	×	R-SIS
Ours	√	√	√	√	M-SIS

6.2. Comparison of Costs

We selected three schemes with similar functionalities to our proposed scheme [20,32,33] for a comparison of computational and communication overhead.

The time comparisons for *MSK* generation, individual user *sk* generation, and *Sig* generation of the three schemes are shown in Table 3. Here, λ represents the security parameter, N denotes the number of ring members, and T_1 represents the average time for the *TrapGen* algorithm. Because [33] is not identity-based, there is no such time overhead. For this part, it is represented by "/". T_2 represents the average time for the *SamplePre* algorithm. T_3 represents the average time for polynomial modular multiplication. T_4 represents the average time for scalar multiplication. Table 3 presents a comparative analysis of the time overhead, individual user *sk* generation time, and signature generation time for the four schemes. We ignored less time-consuming procedures like hash functions and matrix additions in favor of concentrating mostly on computationally demanding operations.

Table 3. Comparison of time costs.

Scheme	MSK-Cost	Ext-Cost	Sig-Cost
[20]	T_1	T_2	$(2N+1)T_4$
[32]	T_1	$T_1 + mT_2$	$2(N+1)T_4$
[33]	/	T_4	$(2N+1)T_4$
Ours	T_1	T_2	$3T_3 + (2N-1)T_4$

Table 4 compares the communication overhead of three schemes in terms of private key size and signature size. Our private key is generated using the *SamplePre* algorithm and is an m-dimensional vector multiplied by the polynomial dimension d. The private key in [20] is generated using *BasisDel* and *SamplePre*, resulting in an m-dimensional vector. The scheme in [33] uses the *SampleDom* algorithm to generate the private key, with the size being the same as in [20]. In our scheme, l_i in the signature is a d-dimensional vector with values in the range $\{-1, 0, 1\}$, so its size is $d \log 3$. The vectors $\boldsymbol{z}$ and $\boldsymbol{\tau}$ are m-dimensional and n-dimensional vectors, respectively, and, since the scheme is based on the M-SIS hard problem, they need to be multiplied by the polynomial dimension d. Although this makes our scheme appear larger in size compared to other schemes, the values of m and n in our scheme are very small, so the signature size is smaller compared to other schemes.

Table 4. Comparison of communication costs.

Scheme	$\lvert sk \rvert$	$\lvert Sig \rvert$
[20]	$m \log q$	$(Nm + N + n) \log q$
[32]	$m \cdot 2^{\lambda} \log 3$	$Nm \cdot 2^{\lambda} \log q + 2^{\lambda} \log 3$
[33]	$m \log q$	$(mN + n) \log q$
Ours	$md \log q$	$Nd \log 3 + (m+n)d \log q$

We set the parameters $d = 128$ and $q = 2^{32} = 4294967296$. In our signature scheme, $|l_i| = (d \log 3)/8$ *bytes*, $|\mathbf{z}| = (md \log q)/8$ *bytes*, and $|\boldsymbol{\tau}| = (nd \log q)/8$ *bytes*. Here are the estimated signature sizes for this scheme with different numbers of ring members based on the parameters in [23].

In Table 5, we provide the sizes of the proposed ring signature with the increase in ring size N, as well as the sizes of responses l_i. "Sig Size" denotes the sizes of signature, while the "Size of $(l_1, \cdots, l_N)$" shows the sizes of the hash values in the ring signatures. Figure 4 shows the increasing trend of communication costs with the ring scale. We can observe that, although the signature size increases linearly with the number of ring members N, the size of $\mathbf{z}$ in the signature does not change significantly. When the number of ring members $N \leq 64$, the signature size does not change much, and it is no more than 13 KB even when $N = 64$. The size is mainly affected by the response $\mathbf{z}$. Therefore, the signature size is mainly related to the number of l_i values. Since the l_i values are very small, the signature size does not change much as the number of ring members increases. When N reaches 128, the signature size is just under 15 KB. When N grows to 2048 and the parameters n and m are chosen to be larger, the signature size is 62.72 KB, which is still acceptable for most application scenarios.

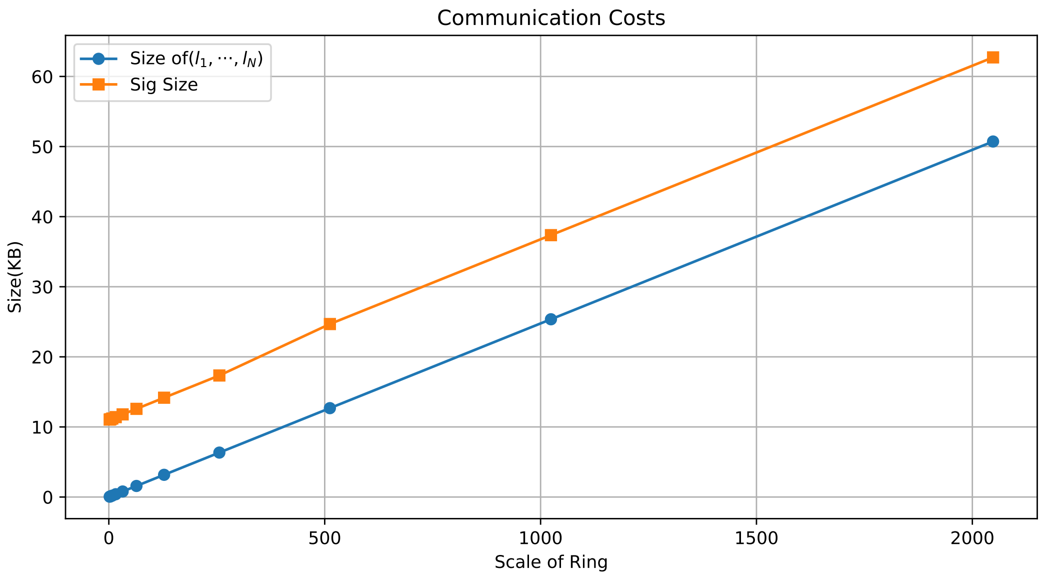

Figure 4. Communication costs with numbers of ring members.

Table 5. Communication costs (KB).

N	n	m	Sig Size	Size of $(l_1, \cdots, l_N)$
2	7	15	11.05	0.05
4	7	15	11.10	0.10
8	7	15	11.20	0.20
16	7	15	11.40	0.40
32	7	15	11.79	0.79
64	7	15	12.58	1.58
128	7	15	14.17	3.17
256	7	15	17.34	6.34
512	8	16	24.68	12.68
1024	8	16	37.36	25.36
2048	8	16	62.72	50.72

7. Identity-Based Threshold Linkable Dual-Ring Signature

To further enhance threshold functionality, we adapt the threshold technique from [34] into our scheme, resulting in an identity-based threshold linkable dual-ring signature scheme (IB-TLDRS). Since most steps of this structure are similar to the previous scheme, we focus on the different steps.

- **IB-TLDRS.Setup**: Same as the setup process in Algorithm 1, except setting a threshold t.
- **IB-TLDRS.KeyExt**: Same as Algorithm 2.
- **IB-TLDRS.Sign**: Same as Algorithm 3.
- **IB-TLDRS.Combine**: A new algorithm required to be added in. The signer sends the generated valid signature Sig to the **IB-TLDRS.Combine** algorithm, which then combines it into a set $(\boldsymbol{\mu}, Sig_0, Sig_1, \cdots, Sig_k, W)$ and sends it to the verification algorithm **IB-TLDRS.Verify**.
- **IB-TLDRS.Link**: Same as Algorithm 5.
- **IB-TLDRS.Verify**: Input $(pp, W, \boldsymbol{\mu}, Sig_0, \cdots, Sig_k)$; after parsing and verifying the signature, the verifier retrieves the successfully verified signature tag in Γ. If it is not in $\Gamma = (\boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_k)$, the tag is added. Finally, if $|\Gamma| > t$, the output is 1; otherwise, the output is 0.

Specifications. Through the above method, we can obtain a new scheme with threshold functionality. For the new scheme, we only need a third party to perform the **IB-LDRS.Combine** algorithm after the signing procedure is completed. In the **Verify** algorithm, it is necessary to first verify the correctness of the signature before checking whether the threshold requirement is met.

Security Analysis. Adding threshold functionality does not affect the security. The threshold functionality mainly relies on the tags in the signature. We have already proven the linkability and nonslanderability, which ensure the security of the tags. This also demonstrates that the scheme can still ensure its security after incorporating the threshold functionality.

8. Conclusions and Future Work

Based on the lattice-based M-SIS assumption, this work constructs an efficient identity-based linkable dual-ring signature scheme, with its threshold extension additionally. The proposed scheme leverages the benefits of dual-ring signatures, which can reduce signature size effectively, especially when the number of ring members is not very large compared to other logarithmic (linkable) ring signatures. Moreover, our scheme, based on identity, simplifies key management processes, reduces computational and communication costs, and offers enhanced security in linkability compared to existing linkable ring signature schemes. Our ring signature is proved to be anonymous, unforgeable, linkable, and nonslanderable in the random oracle model. The research data further show that this work achieves a smaller signature size compared to prior schemes, effectively decreasing storage costs, even though our signature size scales linearly with the number of ring members. Finally, a threshold extension is given as an additional scheme with specifications and security analysis. Although the signature size in this scheme is very small, it increases linearly with the increase in the number of ring members. Therefore, we consider research on the construction of the logarithmic ring signature from lattices as future work.

Author Contributions: Conceptualization, W.G. and H.Y.; methodology, W.G. and H.Y.; software, B.Q.; validation, B.Q., X.D. and Z.Z.; formal analysis, W.G.; investigation, J.Z.; resources, W.G. and B.Q.; writing—original draft preparation, W.G., H.Y. and J.Z.; writing—review and editing, W.G. and H.Y.; visualization, X.D.; supervision, Z.Z.; project administration, B.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grant Nos. 62002288, 62372370, and 62102299, the Key Research and Development Program of Shaanxi (No. 2023-YBGY-015), the Henan Key Laboratory of Network Cryptography Technology under No. LNCT2022-A05, and the Youth Innovation Team of Shaanxi Universities (No. 23JP160).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rivest, R.L.; Shamir, A.; Tauman, Y. How to Leak a Secret: Theory and Applications of Ring Signatures. In *Essays in Memory of Shimon Even*; Springer: Berlin/Heidelberg, Germany, 2001.
2. Li, X.; Mei, Y.; Gong, J.; Xiang, F.; Sun, Z. A Blockchain Privacy Protection Scheme Based on Ring Signature. *IEEE Access* **2020**, *8*, 76765–76772. [CrossRef]
3. Wang, L.; Peng, C.; Tan, W. Secure Ring Signature Scheme for Privacy-Preserving Blockchain. *Entropy* **2023**, *25*, 1334. [CrossRef] [PubMed]
4. van Saberhagen, N. CryptoNote v 2.0. 2013. Available online: https://api.semanticscholar.org/CorpusID:2711472 (accessed on 22 October 2024).
5. Liu, J.K.; Wei, V.K.; Wong, D.S. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract). In *Information Security and Privacy, Proceedings of the 9th Australasian Conference, ACISP 2004, Sydney, Australia, 13–15 July 2004*; Springer: Berlin/Heidelberg, Germany, 2004.
6. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–22 August 1984; Springer: Berlin/Heidelberg, Germany, 1984.
7. Chow, S.S.; Yiu, S.M.; Hui, L.C. Efficient Identity Based Ring Signature. In *Applied Cryptography and Network Security, Proceedings of the Third International Conference, ACNS 2005, New York, NY, USA, 7–10 June 2005*; Springer: Berlin/Heidelberg, Germany, 2005.
8. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil Pairing. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2001; Springer: Berlin/Heidelberg, Germany, 2001.
9. Zhang, J. An Efficient Identity-Based Ring Signature Scheme and Its Extension. In *Communication Systems and Applications*; Springer: Berlin/Heidelberg, Germany, 2007.
10. Deng, L.; Jiang, Y.; Ning, B. Identity-Based Linkable Ring Signature Scheme. *IEEE Access* **2019**, *7*, 153969–153976. [CrossRef]
11. Nassurdine, M.; Zhang, H.; Zhang, F. Identity Based Linkable Ring Signature with Logarithmic Size. In *Information Security and Cryptology, Proceedings of the 17th International Conference, Inscrypt 2021, Virtual Event, 12–14 August 2021*; Springer: Berlin/Heidelberg, Germany, 2021.
12. Odoom, J.; Huang, X.; Wang, L. Stateless forward-secure key-insulated linkable ring signature scheme in ID-based setting. *J. Syst. Archit.* **2022**, *129*, 102600. [CrossRef]
13. Wang, S.; Zheng, S.; Zhan, T. Identity-Based Linkable and Convertible Ring Signature: Identity-Based Linkable and Convertible Ring Signature. *J. Electron. Inf. Technol.* **2011**, *30*, 995–998. [CrossRef]
14. Singh, P.; Hasabnis, A.R.; Rajpoot, S.; Singh, P. Quantum Attacks on Public Cryptosystems. In Proceedings of the 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 14–16 September 2023; pp. 36–41. [CrossRef]
15. Zhang, Z.; Wu, W.; Sui, H.; Wang, B. Quantum Attacks on Type-3 Generalized Feistel Scheme and Unbalanced Feistel Scheme with Expanding Functions. *Chin. J. Electron.* **2023**, *32*, 209–216. [CrossRef]
16. Bernstein, D.J. Introduction to Post-Quantum Cryptography. 2009. Available online: https://api.semanticscholar.org/CorpusID:61401925 (accessed on 22 October 2024).
17. NIST. NIST Selects Four Post-Quantum Cryptography Algorithms for Standardization [Press Release]. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography (accessed on 22 October 2024).

18. Alberto Torres, W.A.; Steinfeld, R.; Sakzad, A.; Liu, J.K.; Kuchta, V.; Bhattacharjee, N.; Au, M.H.; Cheng, J. Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions in Blockchain (Lattice RingCT v1.0). In *Information Security and Privacy, Proceedings of the 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, 11–13 July 2018*; Springer International Publishing: Cham, Switzerland, 2018.
19. Le, H.Q.; Bay, V.; Dung, H.D.; Willy, S.; Ngoc-Thao, L.; Kazuhide, F.; Kiyomoto, S. Identity-Based Linkable Ring Signatures from Lattices. *IEEE Access* **2021**, *9*, 84739–84755. [CrossRef]
20. Tang, Y.; Xia, F.; Ye, Q.; Wang, M.; Mu, R.; Zhang, X. Identity-Based Linkable Ring Signature on Lattice. *J. Cryptologic Res.* **2021**, *8*, 232–247. [CrossRef]
21. Dodis, Y.; Kiayias, A.; Nicolosi, A.; Shoup, V. Anonymous Identification in Ad Hoc Groups. In *Advances in Cryptology-EUROCRYPT 2004, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004*; Springer: Berlin/Heidelberg, Germany, 2004.
22. Groth, J.; Kohlweiss, M. One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin. In Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 26–30 April 2015; Springer: Berlin/Heidelberg, Germany, 2015.
23. Yuen, T.H.; Esgin, M.F.; Liu, J.K.; Au, M.H.; Ding, Z. DualRing: Generic Construction of Ring Signatures with Efficient Instantiations. In Proceedings of the Annual International Cryptology Conference, Virtual Event, 16–20 August 2021; Springer International Publishing: Cham, Switzerland, 2021.
24. Feng, M.; Lin, C.; Wu, W.; He, D. SM2-DualRing: Efficient SM2-based ring signature schemes with logarithmic size. *Comput. Stand. Interfaces* **2024**, *87*, 103763. [CrossRef]
25. Koo, Z.; No, J.S.; Kim, Y.S. Reduction From Module-SIS to Ring-SIS Under Norm Constraint of Ring-SIS. *IEEE Access* **2020**, *8*, 140998–141006. [CrossRef]
26. Jeong, I.R.; Kwon, J.O.; Lee, D.H. Analysis of Revocable-iff-Linked Ring Signature Scheme. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2009**, *92*, 322–325. [CrossRef]
27. Lyubashevsky, V. Lattice Signatures Without Trapdoors. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012; Springer: Berlin/Heidelberg, Germany, 2012.
28. Pointcheval, D.; Stern, J. Security Arguments for Digital Signatures and Blind Signatures. *J. Cryptol.* **2015**, *13*, 361–396. [CrossRef]
29. Abe, M.; Ohkubo, M.; Suzuki, K. 1-out-of-n Signatures from a Variety of Keys. In *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*; Springer: Berlin/Heidelberg, Germany, 2002.
30. Wang, J. Ring Signature and Identity-Based Ring Signature from Lattice Basis Delegation. *IACR Cryptol. ePrint Arch.* **2010**, *2010*, 378.
31. Hu, M.; Zhang, W.; Liu, Z. An Improved Lattice-Based Ring Signature with Unclaimable Anonymity in the Standard Model. *Comput. J.* **2022**, *66*, 2542–2553. [CrossRef]
32. Cao, C.; You, L.; Hu, G. A Novel Linkable Ring Signature on Ideal Lattices. *Entropy* **2023**, *25*, 237. [CrossRef] [PubMed]
33. Baum, C.; Lin, H.; Oechsner, S. Towards Practical Lattice-Based One-Time Linkable Ring Signatures. In Proceedings of the International Conference on Information and Communications Security, Lille, France, 29–31 October 2018; Springer International Publishing: Cham, Switzerland, 2018.
34. Aranha, D.F.; Hall-Andersen, M.; Nitulescu, A.; Pagnin, E.; Yakoubov, S. Count Me In! Extendability for Threshold Ring Signatures. In Proceedings of the 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, 8–11 March 2022.