Hindawi
Security and Communication Networks
Volume 2021, Article ID 9992414, 17 pages
https://doi.org/10.1155/2021/9992414

Research Article

Identity-Based Linkable Ring Signature on NTRU Lattice

Yongli Tang,[1] **Feifei Xia**,[1] **Qing Ye**,[1] **Mengyao Wang**,[1] **Ruijie Mu**,[1] **and Xiaohang Zhang**[2]

[1]*School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China*
[2]*Henan College of Industry & Information Technology, Jiaozuo 454000, China*

Correspondence should be addressed to Qing Ye; yeqing@hpu.edu.cn

Received 22 March 2021; Revised 4 August 2021; Accepted 16 August 2021; Published 16 September 2021

Academic Editor: Mohamed Amine Ferrag

Although most existing linkable ring signature schemes on lattice can effectively resist quantum attacks, they still have the disadvantages of excessive time and storage overhead. This paper constructs an identity-based linkable ring signature (LRS) scheme over NTRU lattice by employing the technologies of trapdoor generation and rejection sampling. The security of this scheme relies on the small integer solution (SIS) problem on NTRU lattice. We prove that this scheme has unconditional anonymity, unforgeability, and linkability under the random oracle model (ROM). Through the performance analysis, this scheme has a shorter size of public/private keys, and when the number of ring members is small (such as $N \leq 8$), this scheme has a shorter signature size compared with other existing latest lattice-based LRS schemes. The computational efficiency of signature has also been further improved since it only involves multiplication in the polynomial ring and modular operations of small integers. Finally, we implemented our scheme and other similar schemes, and it is shown that the time for the signature generation and verification of this scheme decreases roughly by 44.951% and 33.503%, respectively.

1. Introduction

With the rise in cryptocurrencies represented by Bitcoin in recent years, blockchain [1] technology has attracted widespread attention. It is increasingly being used for electronic voting, medical information sharing, and intellectual property. However, the transmission and storage of data on the blockchain are publicly visible and anyone can access it. Only through the approach of "pseudo-anonymity" to protect the privacy of both parties in the transaction cannot satisfy complete privacy protection requirements. Also, researchers pointed out that ring signature is one of the approaches which are expected to solve this problem.

Rivest et al. [2] first proposed ring signature at Asiacrypt 2001. In the ring signature scheme, any ring member can produce a signature by using their own private key and the public keys of all members. The verifier can only identify whether the signature is produced by a ring member and cannot determine which specific member generated the signature. Therefore, ring signature is anonymous and can be widely used in electronic cash, electronic voting, etc. Most ring signatures are designed based upon conventional public key infrastructure (PKI). Under the PKI system, the user's identity information and public key are bound together by a digital certificate. If the number of ring members is excessive, the management, storage, and verification of certificates will occupy a large amount of system resources and become the bottleneck of the whole system. Shamir [3] constructed an identity-based cryptography to tackle the problems mentioned, in which the public key is calculated by the key generation center (KGC) according to the user's identity information (e.g., ID number, e-mail address, and so on). Then, the user's private key is obtained based on the system master secret key (MSK) and public key. KGC no longer needs to manage a series of certificates. Since it improves the computation performance of the system and avoids certificate management, identity-based cryptography has gained wide attention in these years.

In 2002, Zhang and Kim [4] proposed the first identity-based ring signature. Subsequently, many such schemes [5–8] have been proposed. Liu et al. [9] incorporated linkability into ordinary ring signatures and constructed the

first LRS scheme based upon the discrete logarithm problem (DLP) in 2004. Besides having the anonymity and unforgeability of ordinary ring signatures, LRS can also detect whether the user has completed two or more signatures with the identical private key. When applied to the blockchain, it effectively verifies whether users have double-spending problems while protecting the privacy of blockchain users. Currently, LRS performs a pivotal role in the application domains such as cryptocurrency, electronic elections, and electronic cash [10–12]. Au et al. [13] first designed an identity-based constant-size LRS scheme in 2013, and its security is proved based upon DLP under the ROM. The LRS with unconditional anonymity was given by Liu et al. [14] in the same year, which further improved the security of LRS and made up for the weaknesses of linkability and strong anonymity in ring signature that cannot be realized simultaneously. In 2019, Deng et al. [15] constructed the most efficient identity-based LRS scheme, which requires only seven pairing operations in signature generation and verification. The above schemes are mainly founded on classical number theory problems (e.g., the large integer decomposition [16] and the finite field discrete logarithm problem [17]). These cryptosystems will be breached in polynomial time due to the threats brought by the attacks of quantum computers. Shor [18] pointed that the scheme constructed on the classical number theory problems will no longer be safe in that it cannot effectively resist quantum attacks. If the ring signature is still designed based upon the classical number theory problems, the security of ring signature will be difficult to guarantee in the quantum era.

In the course of searching for the replacement of traditional public key cryptography, the public key cryptosystem on lattice is becoming a prominent candidate for anti-quantum attack cryptographic algorithm. Besides, since it mostly involves matrix-vector multiplication and polynomial-polynomial multiplication operations on lattice, compared with the schemes designed on classical number theory problems, the new lattice-based cryptosystem has attracted extensive attention because it has better asymptotic efficiency and parallelization and is resistant to quantum attacks and other merits. In 2010, Rückert [19] first designed an identity-based ring signature over lattice. In 2012, Tian et al. [20] gave an efficient identity-based ring signature on lattice, and the safety is proved under choosing subring and adaptive chosen-message attack, which to some extent could improve the security of this scheme. However, the computational efficiency of the signature is low owing to the large length of the key and signature. Then, other identity-based cryptosystems [21–23] were proposed. In 2017, Yang et al. [24] constructed the first LRS on lattice based on the accumulator, zero-knowledge proof, and weak pseudorandom function. In 2018, Torres et al. [25] designed a lattice-based LRS with unconditional anonymity, and the security of this scheme relies on the hardness of the SIS [26] problem. The signature generation is more efficient because the rejection sampling algorithm is adopted, and this scheme is applied to the confidentiality agreement for promotion. That same year, Baum et al. [27] produced a more efficient LRS scheme based upon the collision-resistance hash function on lattice, whose security is based on the problem of Module-LWE and Module-SIS. In 2020, Beullens et al. [28] gave the logarithmic LRS from isogeny and lattice assumptions; the length of the signature has a logarithmic relationship with the number of ring members. But they generally have the disadvantages of high communication costs and lower computation performance. However, the NTRU lattice is a particular lattice based on the polynomial ring, which attracts wide attention because the signature is designed on the NTRU lattice cryptosystem requiring a shorter size of key and signature, and the efficiency of computational can be improved greatly. In 2019, Lu et al. [29] designed the first practical and efficient LRS based upon the chameleon hash plus (CH+) function on NTRU lattice. The signature length of this scheme is short, and the efficiency of signature generation is further promoted compared with other similar lattice-based schemes.

1.1. Our Construction. To decrease the signature length and further promote the efficiency of computation for the LRS scheme, we constructed an identity-based LRS scheme over NTRU lattice, and the architecture of the proposed scheme is shown in Figure 1. Our main contributions are as follows. (1) Combining NTRU lattice with identity-based ring signature and adopting the compact Gaussian sampler (CGS) algorithm and rejection sampling techniques to design an identity-based LRS. (2) We proved that the scheme proposed in this paper has unconditional anonymity, unforgeability, and linkability under the ROM. The unforgeability of this scheme relies on the SIS problem over NTRU lattice. (3) The performance analyses in two sides of time costs and storage overhead are provided in detail. It is indicated that this scheme has a smaller size of key and signature, and the computational efficiency of signature generation and verification has also been further increased through the comparison with other similar schemes. Finally, we implemented our scheme and other schemes [15, 28, 29], and it is shown that the time for signature generation and verification of this scheme decreases roughly by 44.951% and 33.503%, respectively, compared with other three existing latest LRS schemes [15, 28, 29]. Compared with latest lattice-based LRS schemes [28, 29], the proposed scheme has the smallest public and private key size and also has the smallest signature size when $N \leq 8$.

1.2. Organization. The remainder of this paper is structured as follows. At first, we introduce the symbols, NTRU lattice, the NTRU-SIS problem, and some algorithms in Section 2. Then, we introduce the definition and security model of identity-based LRS in Section 3. In Section 4, we introduce the proposed scheme. The security analysis of this scheme is shown in Section 5. In Section 6, we discuss how initial parameters are selected and point at the next research directions. In Section 7, we make a detailed performance comparison with other three existing schemes [15, 28, 29]. Finally, we present the experimental results of this scheme and related schemes in Section 8.

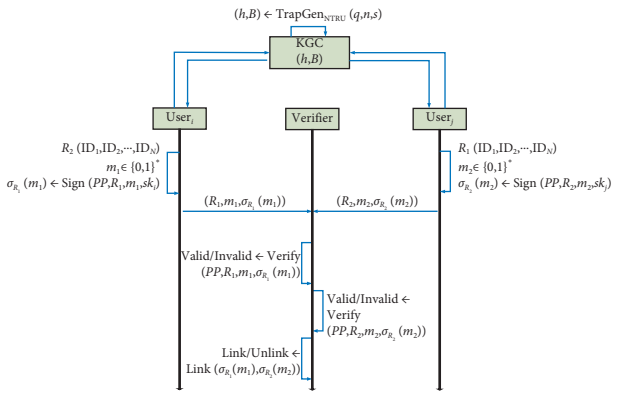

Figure 1: Identity-based linkable ring signature on NTRU lattice.

2. Preliminaries

2.1. Symbol Definition. For the convenience of presentation, the descriptions of the used notations are illustrated in Table 1.

Besides the symbols in Table 1, this paper also uses symbols such as $\widetilde{O}$, ω, which are commonly used symbols for computation complexity.

2.2. Related Definitions of NTRU Lattice

Definition 1 (convolutional polynomial ring). Let ring $\mathbf{R} = (\mathbb{Z}[x]/\langle x^n + 1\rangle)$; when the addition operation on $\mathbf{R}$ remains unchanged and the multiplication operation can be replaced by a convolution operation, then $\mathbf{R}$ is called a convolution polynomial ring. Therefore, given a prime number q and a modular convolution polynomial ring $\mathbf{R}_q = (\mathbf{R}/q\mathbf{R})$.

Let $f = \sum_{i=0}^{n-1} f_i x^i$, $g = \sum_{i=0}^{n-1} g_i x^i \in \mathbf{R}_q$; then, the two operations on $\mathbf{R}_q$ are defined as follows:

(1) Addition operation $+$:
$f + g = g = \sum_{i=0}^{n-1} (f_i + g_i) x^i \bmod q \in \mathbf{R}_q$.

(2) Convolution operation $*$:
$f * g = f * g \bmod (x^n + 1) \in \mathbf{R}_q$.

Definition 2 (anti-circular matrices). Let polynomial $f \in \mathbf{R}_q$, and the coefficient vector of the polynomial f is $(f_0, f_1, \ldots, f_{n-1})$; then, the coefficient vector of the polynomial $x \cdot f$ is $(-f_{n-1}, f_0, \ldots, f_{n-2})$. By analogy, the coefficient vector of the polynomial $x^{n-1} \cdot f$ is $(-f_1, -f_2, \ldots, f_0)$. That is, the anti-circular matrix $\mathbf{A}_n(f)$ is composed of n polynomial vectors formed by successive cyclic shifts of the polynomial $f \in \mathbf{R}_q$. So, the anti-circular matrix $\mathbf{A}_n(f)$ can be expressed as a vector. The anti-circular matrix formed by polynomial f is as follows:

$$\mathbf{A}_n(f) = \begin{bmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ -f_{n-1} & f_0 & \cdots & f_{n-2} \\ \vdots & \vdots & \cdots & \vdots \\ -f_1 & -f_2 & \cdots & f_0 \end{bmatrix} = \begin{bmatrix} f \\ x * f \\ \vdots \\ x^{n-1} * f \end{bmatrix}. \quad (1)$$

Definition 3 (NTRU lattice). n is a security parameter and is a power-of-two integer; let a prime $q \geq 2$ and polynomials $f, g \in \mathbf{R}_q$, $h = g * f^{-1} \bmod q$, satisfying f is reversible, where $f^{-1} \in \mathbf{R}_q$. Then, the NTRU lattice related to q and h is as follows:

$$\Lambda_{q,h} = \left\{ (u, v) \in \mathbf{R}^2 | u + v * h = 0 \bmod q \right\}. \quad (2)$$

NTRU lattice $\Lambda_{q,h}$ is a $2n$ dimension full-rank lattice, $A_{q,h} = \begin{pmatrix} -\mathbf{A}_n(h) & \mathbf{I}_n \\ q\mathbf{I}_n & 0_n \end{pmatrix} \in \mathbb{Z}_q^{2n \times 2n}$ is a set of basis matrix of NTRU lattice, and it is uniquely defined by the polynomial $h \in \mathbf{R}_q$. Therefore, the space required to store the basis matrix $\mathbf{A}_{q,h}$ is small. However, in the application of NTRU lattice-based cryptosystem, $\mathbf{A}_{q,h}$ cannot be used as a trapdoor basis because $\mathbf{A}_{q,h}$ has a very large orthogonal defect when the polynomial $h \in \mathbf{R}_q$ is uniformly distributed.

Definition 4 (discrete Gaussian distribution). For any $\sigma > 0$ and vector $\mathbf{c} \in \mathbf{R}^m$, the discrete Gaussian distribution centered on vector $\mathbf{c}$ over the lattice Λ is described as follows:

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\Lambda)} = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})}, \quad \forall \mathbf{x} \in \Lambda, \quad (3)$$

where $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / \sigma^2)$.

When $\mathbf{c} = 0$, the Gaussian distribution on $\mathbf{R}^m$ and the discrete distribution on lattice Λ can also be defined as D_σ^m and $D_{\Lambda,\sigma}^m$, respectively.

Lemma 1. *Given any parameter $\sigma > 0$ and positive integer m, the following formulas hold [30]:*

(1) $\Pr[\mathbf{x}, \longleftarrow, D_\sigma^1 : \|\mathbf{x}\| > \omega(\sigma\sqrt{\log m})] = 2^{-\omega(\log m)}$.

(2) For any vector $\mathbf{x} \in \mathbb{Z}^m$ and $\sigma \geq \sqrt{\log 3\ m}$, there is $D_\sigma^m(\mathbf{x}) \leq 2^{-m+1}$.

(3) $\Pr[\mathbf{x}, \longleftarrow, D_\sigma^m : \|\mathbf{x}\| > 2\sigma\sqrt{m}] < 2^{-m}$.

2.3. The NTRU-SIS Problems

Definition 5 (the SIS problem on NTRU lattice, NTRU-SIS). Given parameters n, m, q, polynomial $h = g * f^{-1} \bmod q \in \mathbf{R}_q$, and a real number $\beta > 0$, the NTRU-SIS problem is defined as follows: finding two non-zero small polynomials $(u, v) \in \mathbf{R}_q^2$ satisfying $u + v * h = 0 \bmod q$ and $\|u\|, \|v\| \leq \beta$.

NTRU-SIS Assumption. Given system linear equations of modulo q, without knowing the trapdoor, the advantage of finding two non-zero small polynomials $(u, v) \in \mathbf{R}_q^2$ that meet $u + v * h = 0 \bmod q$ and $\|u\|, \|v\| \leq \beta$ can be neglected for any probabilistic polynomial time (PPT) algorithm.

2.4. Related Algorithms

Definition 6 (trapdoor generation algorithm on NTRU lattice). Given integers n and k, where $k > 0$ and $n = 2^k$, and a prime $q = 1 \bmod 2n$, let a parameter $s = 1.17\sqrt{q/2n}$ and

Table 1: Symbol description.

Notations	Explanation
$\mathbf{R}$	Polynomial ring $\mathbf{R} = (\mathbb{Z}[x]/\langle x^n + 1\rangle)$
$\mathbb{Z}$	Integer set
$\mathbf{R}_q$	Integer modulo q polynomial ring $\mathbf{R}_q = (\mathbb{Z}_q[x]/\langle x^n + 1\rangle)$
s^l	First l bits of string s
$\mathbf{A}$	Matrices
$\mathbb{Z}_q^n$	n-dimensional vector of modulo q residue class ring
$\mathbb{Z}_q^{n\times m}$	$n \times m$-dimensional matrix space of modulo q residue class ring
$\mathbb{Z}^m$	m-dimensional integer vector space
$\mathbf{x}$	Vectors, assumed to be in column form
$\mathbf{R}^m$	m-dimensional real vector space
$\mathbf{x} \longleftarrow D$	Chooses vector $\mathbf{x}$ from probability distribution D
$\|\mathbf{x}\|$	Euclidean norm of vector $\mathbf{x}$
$\mathbb{R}$	Set of real numbers
$\mathrm{negl}(n)$	The negligible function of n

$f, g, F, G \in \mathbf{R}$ satisfying $f \cdot F - g \cdot G = q$. The PPT algorithm $\mathrm{TrapGen}_{\mathrm{NTRU}}(q, n, s)$ [31] can output a polynomial $h = g * f^{-1} \bmod q$ and a set of short basis $\mathbf{B}_{f,g} = \begin{bmatrix} \mathbf{A}_n(g) & -\mathbf{A}_n(f) \\ \mathbf{A}_n(G) & -\mathbf{A}_n(F) \end{bmatrix} \in \mathbb{Z}_q^{2n\times 2n}$ on NTRU lattice $\Lambda_{h,q}$.

Definition 7 (discrete Gaussian sampling function). In 2015, Lyubasevsky and Prest [32] proposed the compact Gaussian sampler (CGS) algorithm that can quickly implement discrete Gaussian distribution sampling on NTRU lattice.

Theorem 1. *This is a more efficient polynomial time algorithm $CGS(\mathbf{B}, \sigma, \mathbf{c})$ [32]: on inputting a lattice basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, Gaussian parameter σ, and center vector $\mathbf{c} \in \mathbb{Z}^m$, the algorithm $CGS(\mathbf{B}, \sigma, \mathbf{c})$ can output the sampling $\mathbf{z}$ on distribution $D_{\Lambda(\mathbf{B}),\sigma,\mathbf{c}}$.*

Lemma 2. *Let a parameter $\sigma \geq \|\mathbf{B}\| \cdot (1/\pi)\sqrt{(1/2)\ln(2 + (2/\varepsilon))}$, where $\varepsilon < (2^{-\lambda}/4n)$, $\log_2 n < \lambda < (qn/2)$, such that the statistical distance between the output of $CGS(\mathbf{B}, \sigma, \mathbf{c})$ and the distribution $D_{\Lambda(\mathbf{B}),\sigma,\mathbf{c}}$ is no more than $2^{-\lambda}$.*

Definition 8 (rejection sampling). In 2012, Lyubasevsky [30] proposed the rejection sampling and, based on this technology, first designed the signature scheme without trapdoor on the lattice. This technology can be applied to the signature system, which obtains the signature with a definite probability and makes the distribution of the signature and private key separate from one another so that the signature private key can be effectively prevented from leaking. The conclusions are as follows.

Lemma 3. *For any $\mathbf{v} \in \mathbb{Z}^m$, $\sigma = \omega(\|\mathbf{v}\|\sqrt{\log m})$, we have* $\Pr[((D_\sigma^m(\mathbf{z}))/(D_{\mathbf{v},\sigma}^m(\mathbf{z}))) = O(1) : \mathbf{z} \longleftarrow D_\sigma^m] = 1 - 2^{-\omega(\log m)}$.

Theorem 2. *Let $V = \{\mathbf{v} \in \mathbb{Z}^m : \|\mathbf{v}\| < t\}$, $\sigma = \omega(t\sqrt{\log m})$, h: $V \longrightarrow \mathbb{R}$ be a probability distribution; for any constant $M = O(1)$, the statistical distance of output distribution between Algorithm 1 and Algorithm 2 is less than $(2^{-\omega(\log m)}/M)$.*

Algorithm 1. $v \longleftarrow h; \mathbf{z} \longleftarrow D_{\mathbf{v},\sigma}^m$: output $(\mathbf{z}, v)$ with the probability of $\min((D_\sigma^m(\mathbf{z})/MD_{v,\sigma}^m(\mathbf{z})), 1)$.

Algorithm 2. $v \longleftarrow h; \mathbf{z} \longleftarrow D_\sigma^m$: output $(\mathbf{z}, v)$ with the probability of $(1/M)$.

Furthermore, the output probability of Algorithm 1 is at least $1 - (2^{-\omega(\log m)}/M)$.

3. Definition of Identity-Based LRS and Security Model

3.1. Definition of Identity-Based LRS. An identity-based LRS [14, 33] is composed of five PPT algorithms:

(1) $\mathrm{Setup}(1^\lambda, 1^N)$: it inputs a security parameter λ, the number of ring members N, and returns the public parameters PP, and system master private key MSK.

(2) $\mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}_i, \mathrm{MSK})$: it inputs public parameters PP, user identity ID_i, and system master private key MSK and returns a pair of public\private key $(\mathrm{pk}_i, \mathrm{sk}_i)$.

(3) $\mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_i)$: it inputs public parameters PP, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and signature private key sk_i of user $\mathrm{ID}_i \in R$ and outputs the ring signature $\sigma_R(m)$ of user ID_i on a message $m \in \{0,1\}^*$, which contains the linkability tag I.

(4) $\mathrm{Verify}(\mathrm{PP}, R, m, \sigma_R(m))$: it inputs public parameters PP, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and a signature $\sigma_R(m)$; if $\sigma_R(m)$ is valid, the verifier returns "valid"; otherwise, it outputs "invalid."

(5) $\mathrm{Link}(\sigma_{R_1}(m_1), \sigma_{R_2}(m_2))$: it inputs two ring signatures $\sigma_{R_1}(m_1)$, $\sigma_{R_2}(m_2)$ and verifies $I_{(1)} \stackrel{?}{=} I_{(2)}$. If equal, the verifier returns "link." It indicates that $\sigma_{R_1}(m_1)$, $\sigma_{R_2}(m_2)$ are produced by identical signer; otherwise, it outputs "unlink."

3.2. Security Model. An identity-based LRS scheme definition of security should meet linkability in addition to correctness, anonymity, and unforgeability of ordinary ring

signatures. Correctness includes verification correctness and linking correctness (refer to Definition 9 for details). Anonymity implies that the attacker is unable to confirm which specific member of the ring generated the signature, and our scheme has strong anonymity, that is, unconditional anonymity (refer to Definition 10 for details). Unforgeability means that the members outside of the ring cannot, instead of the real signer, sign without the signer's private key (refer to Definition 11 for details). Linkability means that users with only one private key cannot give two signatures, which successfully pass the detection of the linking algorithm (refer to Definition 12 for details). This paper is based upon the security model proposed by Liu et al. [14], using a series of games between a challenger C and an adversary A to characterize the security definition of this scheme. The adversary A can call on the random oracle and the oracles RO, CO, SO under the ROM.

(i) Registration oracle: A chooses a random user's identity ID_i to query and C uses the $\mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}_i, \mathrm{MSK})$ algorithm to return the corresponding public key pk_i.

(ii) Corruption oracle: A chooses a random user's identity ID_i to query and C uses the $\mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}_i, \mathrm{MSK})$ algorithm to return the corresponding private key sk_i.

(iii) Signing oracle: A inputs ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and user's identity $\mathrm{ID}_i \in R$ to query, and C gives a valid signature $\sigma_R(m)$ through running $\mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_i)$ algorithm.

Definition 9 (correctness). The correctness of the LRS scheme contains verification correctness and linking correctness simultaneously.

(1) Verification correctness: requires signature $\sigma_R(m)$ generated by users honestly in accordance with the specification; the probability of algorithm $\mathrm{Verify}(\mathrm{PP}, R, m, \sigma_R(m))$ outputting "invalid" is negligible.

(2) Linking correctness: requires two valid signatures $\sigma_{R_1}(m_1)$ and $\sigma_{R_2}(m_2)$ produced with the identical private key for the same signer; the probability of algorithm $\mathrm{Link}(\sigma_{R_1}(m_1), \sigma_{R_2}(m_2))$ outputting "unlink" is negligible.

The formal definition of the correctness of the LRS is as follows:

$$\Pr\left[\text{"Invalid"} \longleftarrow \mathrm{Verify}(\mathrm{PP}, R, m, \sigma_R(m)) \;\middle|\; \begin{array}{l} \mathrm{MSK}, \mathrm{PP} \longleftarrow \mathrm{Setup}\left(1^\lambda, 1^N\right) \\ \mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}, \mathrm{MSK}) \\ \sigma_R(m) \longleftarrow \mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}) \end{array}\right] \leq \mathrm{negl}(n),$$

$$\Pr\left[\text{"Unlink"} \longleftarrow \mathrm{Link}\left(\sigma_{R_1}(m_1), \sigma_{R_2}(m_2)\right) \;\middle|\; \begin{array}{l} \mathrm{MSK}, \mathrm{PP} \longleftarrow \mathrm{Setup}\left(1^\lambda, 1^N\right) \\ \mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}, \mathrm{MSK}) \\ \sigma_{R_1}(m_1) \longleftarrow \mathrm{Sign}(\mathrm{PP}, R_1, m_1, \mathrm{sk}) \\ \sigma_{R_2}(m_2) \longleftarrow \mathrm{Sign}(\mathrm{PP}, R_2, m_2, \mathrm{sk}) \end{array}\right] \leq \mathrm{negl}(n). \quad (4)$$

3.2.1. Unconditional Anonymity. Even if A possesses unlimited computational resources (with unbounded computation power and time), it can compute the corresponding private key when a public key is given. It is not feasible to distinguish the signer's identity with a probability larger than 1/2. The unconditional anonymity of the LRS scheme is defined by the following game between an adversary A and a challenger C.

(1) Setup: on receiving a security parameter λ and the number of ring members N, C calls the $\mathrm{Setup}(1^\lambda, 1^N)$ algorithm to get the public parameters PP and system master private key MSK and gives the public parameters PP to A.

(2) Query: A is allowed to make adaptive inquiries to above oracles.

(3) Challenge: A inputs ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$ and a message $m^* \in \{0,1\}^*$ and chooses two user identities $\mathrm{ID}_{i_0}, \mathrm{ID}_{i_1} \in R$ for signing queries; C randomly selects a number $b \in \{0,1\}$ and then obtains the signature $\sigma_R(m^*) \longleftarrow \mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_i)$. This signature $\sigma_R(m^*)$ is given to A.

(4) Guess: A gives guess $b^* \in \{0,1\}$. The adversary A wins if the conditions described below are satisfied:

(a) $b^* = b$.

(b) $\mathrm{ID}_{i_0}, \mathrm{ID}_{i_1} \in R$ cannot be input by CO and SO.

The advantage of A is denoted by

$$\mathrm{Adv}_A^{\mathrm{anon}} = \left|\Pr[b^* = b] - \frac{1}{2}\right|. \quad (5)$$

Definition 10 (unconditional anonymity). The LRS scheme is unconditionally anonymous if the advantage $\mathrm{Adv}_A^{\mathrm{anon}} \leq \mathrm{negl}(n)$ for any PPT adversary A.

3.2.2. Unforgeability. The unforgeability of the LRS scheme is defined by the following game between an adversary A and a challenger C.

(1) Setup: given a security parameter λ and the number of ring members N, C calls the Setup$(1^\lambda, 1^N)$ algorithm to get the public parameters PP and system master private key MSK and sends the public parameters PP to A.

(2) Query: A is allowed to make adaptive inquiries to above oracles.

(3) Forge: A gives C a tuple $(m^*, R^*, \sigma_{R^*}(m^*))$. The adversary A wins if the conditions described below are satisfied:

(a) Verify$(m^*, R^*, \sigma_{R^*}(m^*))$ = "Valid."
(b) All of the public keys in R^* are inquiry outputs of RO.
(c) The identity of anyone in R^* has not been input to CO.
(d) $\sigma_{R^*}(m^*)$ is not an inquiry output of SO.

The advantage of A is denoted by

$$\mathrm{Adv}_A^{\mathrm{forge}} = \Pr[A\text{wins the game}]. \quad (6)$$

Definition 11 (unforgeability). The LRS scheme is unforgeable if the advantage $\mathrm{Adv}_A^{\mathrm{forge}} \le \mathrm{negl}(n)$ for any PPT adversary A.

3.2.3. Linkability. The linkability of the LRS scheme is defined by the following game between an adversary A and a challenger C.

(1) Setup: on receiving a security parameter λ and the number of ring members N, C calls the Setup$(1^\lambda, 1^N)$ algorithm to get the public parameters PP and system master private key MSK and gives the public parameters PP to A.

(2) Query: A is allowed to make adaptive inquiries to above oracles.

(3) Forge: A gives C two message-signature pairs $(m_1^*, R_1^*, \sigma_{R_1^*}(m_1^*))$, $(m_2^*, R_2^*, \sigma_{R_2^*}(m_2^*))$, and the two signatures $\sigma_{R_1^*}(m_1^*)$, $\sigma_{R_2^*}(m_2^*)$ contain corresponding two linkability tags $I_{(1)}$, $I_{(2)}$. The adversary A wins if the conditions described below are satisfied:

(a) Verify$(m_i^*, R_i^*, \sigma_{R_i^*}(m_i^*))$ = "Valid" for $i = 1, 2$.
(b) Link$(\sigma_{R_1^*}(m_1^*), \sigma_{R_2^*}(m_2^*))$ = "Unlinked."
(c) All of the public keys in R_i^*, $i \in \{1, 2\}$ are outputs of RO.
(d) A has inquired CO less than two times (that is, A has one private key of users at most).
(e) $\sigma_{R_1^*}(m_1^*)$, $\sigma_{R_2^*}(m_2^*)$ are not inquired outputs of SO.

The advantage of A is denoted by

$$\mathrm{Adv}_A^{\mathrm{link}} = \Pr[A\text{wins the game}]. \quad (7)$$

Definition 12 (linkability). The LRS scheme is linkable if the advantage $\mathrm{Adv}_A^{\mathrm{link}} \le \mathrm{negl}(n)$ for any PPT adversary A.

4. Scheme Construction

To decrease the signature length and promote the computational efficiency of existing LRS schemes. We designed an identity-based LRS scheme over NTRU lattice by employing technologies of trapdoor generation algorithm [31] and rejection sampling [30]. The construction idea of this paper is to introduce identity-based cryptography into the efficient NTRU lattice-based ring signature. During the system setup process, the system's master key uses the trapdoor generation [31] algorithm to obtain NTRU lattice. In the key generation period, the private key is produced based upon the compact Gaussian sampler (CGS) algorithm [32], which effectively improves the speed of user key extraction. In the signature generation phase, through using the rejection sampling [30] technique to generate signature with a certain probability, the distribution of signature and private key is independent of each other, and the computational efficiency of signature is further optimized and improved. The proposed identity-based LRS scheme is as follows:

(1) Setup$(1^\lambda, 1^N)$: by inputting a security parameter λ and the number of ring members N, it sets integer $k > 0$, where integer $n = 2^k$, and chooses a prime number $q = 1 \bmod 2n$, a parameter $s = 1.17\sqrt{q/2n}$, and a Gaussian parameter $\sigma = (117/200\pi) \cdot \sqrt{q \cdot \log_{\exp}(2 + 2/\eta)}$, where $\eta = (2^{-(\lambda+1)}/n)$. It sets polynomial ring $\mathbf{R}_q = (\mathbb{Z}_q[x]/\langle x^n + 1\rangle)$, and KGC obtains the public parameters PP and the system master private key MSK according to the following steps.

(a) KGC uses $\mathrm{TrapGen}_{\mathrm{NTRU}}(q, n, s)$ algorithm to generate a uniform and randomized polynomial $h \in \mathbf{R}_q$ together with a short basis $\mathbf{B} \in \mathbb{Z}_q^{2n\times 2n}$ on lattice $\Lambda_{q,h}$.
(b) Selects two collision-resistance hash functions H_1: $\{0,1\}^* \longrightarrow \mathbb{Z}_q^n$, H_2: $\{0,1\}^* \longrightarrow \{0,1\}^n$.
(c) The system master private key of KGC is MSK = $\mathbf{B}$, and master public key is MPK = h.
(d) Outputs the public parameters PP = (h, H_1, H_2) and keeps the system master private key MSK = $\mathbf{B}$ secret.

(2) KeyGen(PP, ID_i, MSK): given the public parameters PP, user's identity ID_i, and system master private key MSK = $\mathbf{B}$, KGC obtains a pair of public\private key $(\mathrm{pk}_i, \mathrm{sk}_i)$ as follows.

(a) Calculates the public key pk_i is $\mathbf{t}_{i,1} = H_1(\mathrm{ID}_i) \in \mathbb{Z}_q^n$.
(b) Uses CGS sampling algorithm to generate $(s_1, s_2) = (\mathbf{t}_{i,1}, 0) - \mathrm{CGS}(\mathbf{B}, \sigma, (\mathbf{t}_{i,1}, 0)$, then $s_1 + s_2 * h = \mathbf{t}_{i,1}$.
(c) Randomly chooses polynomial vectors $\mathbf{s}_1^*, \mathbf{s}_2^* \longleftarrow D_\sigma^n$ and returns user's private key $\mathrm{sk}_i = (s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*)$.

(3) $\mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_k)$: receives the public parameters PP, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and private key $\mathrm{sk}_k = (s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*)$ of user $\mathrm{ID}_k \in R$. The signing process is as follows:

(a) Calculates $I = \mathbf{t}_{k,1} + t_{k,2} \in \mathbf{R}_q$ as the linkability tag, where $t_{k,2} = \mathbf{s}_1^* + \mathbf{s}_2^* * h \in \mathbf{R}_q$.
(b) Randomly chooses polynomial vectors $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow D_\sigma^n$, $i \in \{1, 2, \ldots, N\}$, and the vectors corresponding to short polynomials in $\mathbf{R}_q$.
(c) Sets $v = H_2(\sum_{i=1}^N \mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h, R, m, I)$.
(d) If $i \neq k$, it sets $\mathbf{z}_{i,1} = \mathbf{y}_{i,1}$, $\mathbf{z}_{i,2} = \mathbf{y}_{i,2}$; if $i = k$, it calculates $\mathbf{z}_i = \begin{pmatrix} \mathbf{z}_{i,1} \\ \mathbf{z}_{i,2} \end{pmatrix} = \begin{pmatrix} s_1 + \mathbf{s}_1^* \\ s_2 + \mathbf{s}_2^* \end{pmatrix} * v + \begin{pmatrix} \mathbf{y}_{i,1} \\ \mathbf{y}_{i,2} \end{pmatrix}$.
(e) By probability $(D_{\widetilde{\sigma}(\sqrt{2n}\sigma)}((\mathbf{z}_{i,1}, \mathbf{z}_{i,2}))/MD_{((s_1+s_1^*)v, (s_2+s_2^*)v), \widetilde{\sigma}(\sqrt{2n}\sigma)((\mathbf{z}_{i,1}, \mathbf{z}_{i,2}))})$, it outputs $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$ as the signature, where $M = O(1)$.

(4) $\mathrm{Verify}(\mathrm{PP}, R, m, \sigma_R(m))$: receives the public parameters PP, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and a ring signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$. For $i \in \{1, 2, \ldots, N\}$, calculates and verifies whether the following conditions hold:

(a) $\|\mathbf{z}_{i,1}\| \le 2\sigma\sqrt{n}$, $\|\mathbf{z}_{i,2}\| \le 2\sigma\sqrt{n}$.
(b) $H_2(\sum_{i=1}^N \mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h - I * v, R, m, I) = v$.

If conditions (a) and (b) are satisfied, the verifier will return "Valid" and then accept the message $m \in \{0,1\}^*$ signed by a member of the ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$; otherwise, it outputs "Invalid."

(5) $\mathrm{Link}(\sigma_{R_1}(m_1), \sigma_{R_2}(m_2))$: on inputting two signatures $\sigma_{R_1}(m_1), \sigma_{R_2}(m_2)$, the verifier does these steps:

Inputs two signatures $\sigma_{R_1}(m_1)$ and $\sigma_{R_2}(m_2)$ and verifies $I_{(1)} \stackrel{?}{=} I_{(2)}$; if $I_{(1)} = I_{(2)}$, it outputs "Link"; otherwise, it outputs "Unlink."

5. Security Analysis

Theorem 3 (correctness). *The proposed identity-based LRS scheme is correct.*

Proof. The proof is given in Appendix A. □

Theorem 4 (unconditional anonymity). *The proposed identity-based LRS scheme is unconditionally anonymous.*

Proof. The proof is given in Appendix B. □

Theorem 5 (unforgeability). *The proposed identity-based LRS scheme is unforgeable under the ROM, if the SIS problem on the NTRU is hard.*

Proof. The proof is given in Appendix C. □

Theorem 6 (linkability). *The proposed identity-based LRS scheme is linkable under the ROM, if the proposed LRS scheme is unforgeable.*

Proof. The proof is given in Appendix D. □

6. Discussion

We now discuss how the initial parameters are selected and point at the future research direction.

6.1. Parameter Selection. The security of the proposed scheme is based on the NTRU-SIS problem, which is defined as follows: finding two non-zero small polynomials $(u, v) \in \mathbf{R}_q^2$ satisfying $u + v * h = 0 \bmod q$ and $\|u\|, \|v\| \le \beta$. This problem can be reduced to the γ-ideal-SVP problem. According to the literature [34, 35], the value of γ measures the hardness of γ-ideal-SVP problem. We use the root-Hermite factor (RHF) γ to analyze the security level of the scheme and set the relevant parameters. According to the literature [34], if a polynomial vector v is found in an n-dimensional lattice Λ and the vector is greater than the n^{th} root of the determinant, then the relative RHF is

$$\gamma = \left(\frac{\|v\|}{\det(\Lambda)^{(1/n)}} \right)^{(1/n)}. \tag{8}$$

If the small-size polynomial vector v is found in the NTRU lattice $\Lambda_{q,h}$, then the relative RHF is

$$\gamma = \left(\frac{2.5\sqrt{n/\pi \exp} \cdot \det\left(\Lambda_{q,h}\right)^{1/2n}}{\|v\|} \right)^{1/2n}. \tag{9}$$

According to the results of literature [35, 36], when the value of γ is approximately 1.007, finding the vector satisfying the condition is at least 80-bits hard. When the value of γ is less than 1.004, finding the vector satisfying the condition is at least 192-bits hard.

The methods of attacking the proposed scheme are mainly attacks on the public keys of ring members and the signatures.

In the proposed scheme, the public key of ring member i is $\mathbf{t}_{i,1} = H_1(\mathrm{ID}_i)$. The attack on $\mathbf{t}_{i,1}$ is to find two non-zero small-size polynomials $(s_1, s_2) \in \mathbf{R}_q^2$ in the NTRU lattice that satisfy $s_1 + s_2 * h = \mathbf{t}_{i,1}$. According to Definition 6, $\|(s_1, s_2)\| \le s\sqrt{2n}$. The value of γ is calculated by (9), and we have $\gamma = (\sqrt{n}/1.368)^{1/2n}$. When $n = 256$, $\gamma \approx 1.0068$, attacking the public key of ring members is at least 80-bits hard, and when $n = 512$, $\gamma \approx 1.0027$, attacking the public key of ring members is at least 192-bits hard.

In the proposed scheme, the signature of ring member i is $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$. The attack on the signature is to find vectors $(\mathbf{z}_{i,1}, \mathbf{z}_{i,2})$ through the verification algorithm without the private key of the ring member i. Then, the value of γ is calculated by formula (8), and we have

$$\gamma = \left(\frac{\left\| \left(\mathbf{z}_{i,1}, \mathbf{z}_{i,2} \right) \right\|}{\det\left(\Lambda_{q,h} \right)^{1/2n}} \right)^{1/2n}. \tag{10}$$

From Lemma 3 and Theorem 2, $\|(\mathbf{z}_{i,1}, \mathbf{z}_{i,2})\| \le \sigma\sqrt{2n}$, and we have

$$\gamma = \left(\frac{\sigma\sqrt{2n}}{\sqrt{q}} \right)^{1/2n}. \tag{11}$$

Table 2: Parameter setting for our scheme.

Parameter	n	k	λ	Security level
Recommended choice	256	8	80	80-bits
	512	9	192	192-bits

The Gaussian parameter is defined as $\sigma = (117/200\pi) \cdot \sqrt{q \cdot \log_{\exp}(2 + (2/\eta))}$, where $\eta = (2^{-(\lambda+1)}/n)$. Through further calculation, the following results can be obtained.

$$\gamma = \left(\frac{\sigma\sqrt{2n}}{\sqrt{q}} \right)^{1/2n} \approx \left(0.1862\sqrt{4n + \frac{8n^2}{2^{-\lambda}}} \right)^{1/2n} \approx \begin{cases} 1.0076, & n = 256, \quad \lambda = 80, \\ 1.0038, & n = 512, \quad \lambda = 192. \end{cases} \tag{12}$$

When $n = 256$, $\gamma \approx 1.0076$, attacking the signature of ring members is at least 80-bits hard, and when $n = 512$, $\gamma \approx 1.0038$, attacking the signature of ring members is at least 192-bits hard. The main parameters of this scheme are defined in Table 2.

6.2. Post-Quantum Security. It is generally believed that the proposed scheme constructed based on the hardness assumption over lattices may provide post-quantum security. On the other hand, the security proof of the proposed scheme is unlikely to carry over to the quantum random oracle model [37] (QROM). We use adaptive programming of the RO H_1 and H_2 in the security proof (Theorems 4–6). This proof technology is inherent to the construction to some extent.

We learned that other construction schemes in the QROM, such as [38, 39], also use a form of RO programming (even if they are not adaptive). As far as we know, though it seems unlikely that the Fiat–Shamir can be proven in the QROM, there are no attacks on the protocols using these proof techniques which are derived from the use of the RO. If the security of the scheme is proved in the QROM, the construction process of the proposed scheme may be subverted. In the next step, we will consider constructing an identity-based LRS on the NTRU lattice under the QROM.

7. Performance Analysis

In this section, we choose three similar schemes to carry out efficiency analysis and comparison with our scheme. They are, respectively, the identity-based LRS scheme based on the bilinear pairings constructed by Deng et al. [15], the logarithmic (linkable) ring signatures on lattice from isogeny and lattice assumptions given by Beullens et al. [28], and the practical lattice-based LRS based upon the chameleon hash plus (CH+) function designed by Lu et al. [29]. We will perform efficiency analysis of our scheme and the other three schemes [15, 28, 29] and mainly focus on two areas: time costs and storage overhead.

The time cost comparison and difficult assumption of the four schemes are listed in Table 3. Comparison terms in Table 3 include signature generation cost, signature verification cost, difficult assumption,and other comparisons. This paper mainly analyzes relatively time-consuming processes such as matrix-vector multiplication and polynomial-polynomial multiplication operation, the pairing operation, and exponentiation operation. The relatively less time-consuming operations such as hash operation, polynomial and matrix addition, and subtraction operation are ignored. In Tables 3 and 4, n represents a positive integer, q is a large prime number, N represents the number of ring members, and k and l are small integers, e.g., $k = 3, l = 4$. $T_{\mathrm{SD}}, T_{\mathrm{RS}}$, respectively, represent the time spent for the discrete Gaussian sampling algorithm and the algorithm rejection sampling run once, and generally, $T_{\mathrm{SD}} > T_{\mathrm{RS}}$. T_{B_p} represents the pairing operation, and $T_{M_{G_1}}, T_{E_{G_2}}$ represent the scalar multiplication operation in additive group G_1 and the exponentiation operation in group G_2, respectively. Besides, T_1 and T_2, respectively, defined the time cost running the polynomial-polynomial multiplication and matrix-vector multiplication operation n times, and generally, $T_2 > T_1$.

The scheme [15] is designed based on DBDH problem, which cannot resist the attack from a quantum computer, while the other three schemes are designed on lattice which can resist quantum computer attack. So, we only list the efficiency of the scheme and no longer compare it with other three schemes. However, in Section 8, the experimental evaluation of scheme [15] will be carried out and compared with our scheme. With respect to signature generation, our scheme mainly uses $2N$ times of the Gaussian sampling algorithm, $N + 1$ times of the polynomial-polynomial multiplication operations, and one-time rejection sampling algorithm. Therefore, the time cost of signature generation is $2nNT_{\mathrm{SD}} + n(N + 1)T_1 + 2nT_{\mathrm{RS}}$. From Table 3, our scheme achieves higher efficiency in signature generation compared with the scheme of [29]. The signature generation time of scheme [28] has logarithmic relationship with the number of ring members N while our scheme has linear relationship with N. When N becomes larger, it is believed that the signature generation time of [28] is superior to

Table 3: Comparison of time costs and difficult assumption.

Scheme	Signature cost	Verification cost	Difficult assumption	Identity-based	Post-quantum
[15]	$4T_{B_P} + (N+1)T_{M_{G_1}}$	$3T_{B_P} + NT_{M_{G_1}} + T_{E_{G_2}}$	DBDH	Yes	No
[28]	$\ln T_{\mathrm{SD}} + 5(k \times n)T_2 \log N$	$2(k \times n)T_2 \log N$	MSIS, MLWE	No	Yes
[29]	$2n(N+1)T_{\mathrm{SD}} + 2n(N+1)T_1$	$2nNT_1$	CH+	No	Yes
Ours	$2nNT_{\mathrm{SD}} + n(N+1)T_1 + 2nT_{RS}$	nNT_1	NTRU-SIS	Yes	Yes

Table 4: Comparison of storage overhead.

Scheme	Public key size	Private key size	Signature size
[15]	$2 \log q$	$2 \log q$	$(4+N)\log q$
[28]	$(k \times l)n \log q$	$\ln \log q$	$(k \times l)n \log q + n \log N$
[29]	$n \log q$	$9n \log q$	$(3N+1)n \log q$
Ours	$n \log q$	$4n \log q$	$(2N+1)n \log q$

that of our scheme. However, when N is small, which scheme has better signature generation efficiency depends on the concrete setting of relevant parameters, e.g., k, l, n, N. In terms of signature verification, our scheme mainly carries out N times of polynomial-polynomial multiplication operations. Therefore, the signature verification time cost of the proposed scheme is nNT_1. Compared with the scheme of [29], the proposed scheme obviously has higher efficiency. The comparison of signature verification is similar to that of signature generation between the scheme of [28] and ours. Moreover, our scheme provides identity-based properties, which effectively avoids the problem of certificate management. However, neither scheme [28] nor scheme [29] has this property. When the number of ring members is large, the management and verification of certificates will take up a lot of system resources and the efficiency of signature and verification will be affected.

The storage overhead comparison of the four schemes is shown in Table 4; comparison terms include the size of public/private key and the signature size. On the size of public/private key, the public key is defined as a n-dimensional vector, and the private key matches the four small polynomials in the ring $\mathbf{R}_q$ in this paper. Therefore, the size of public key is $n \log q$, and the size of private key is $4n \log q$. For scheme [15], the public key and private key are elements in group G_1. Therefore, the public\private key size is $2 \log q$. Regarding scheme [28], the public key is obtained by multiplying two random ring polynomials from $\mathbf{R}_q^l$ and $\mathbf{R}_q^k$, respectively, and the private key is defined as ring polynomials from $\mathbf{R}_q^l$. Therefore, the public\private key size is $(k \times l)n \log q$ and $ln \log q$, respectively. In scheme [29], the public key is defined as a small polynomial in the ring $\mathbf{R}_q$, and the private key matches the nine small polynomials in the ring $\mathbf{R}_q$. Therefore, the size of public key is $n \log q$, and the size of private key is $9n \log q$. According to the comprehensive analysis, scheme [15] has the smallest public and private key size, the public key length of the proposed scheme is much smaller than [28], and the private key of the proposed scheme is significantly smaller than that of [29]. Moreover, when $l > 4$, the length of the private key of this scheme is also smaller than that of scheme [28]. For signature size, the signature generated in this article is $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$. The polynomial vectors $(\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, I)$ correspond to the small-size polynomials in the ring $\mathbf{R}_q$. Therefore, the signature size of our scheme is $(2N+1)n \log q$. After comparison, the signature length of this scheme is smaller than that of scheme [29]. Because the signature size of scheme [28] is logarithmic, when the number of ring members N is large, the signature length of the proposed scheme is larger than that of scheme [28] and needs to be further optimized. Although the signature size of the proposed scheme is longer than that of [15], our scheme is designed based on hardness assumption over lattice and can effectively resist quantum attacks, while the scheme of [15] cannot.

Table 5: Parameter setting for our scheme.

Parameter	Type	n	k	q	Security level
Recommended choice	I	256	8	2^{32}	80-bits
	II	512	9	2^{33}	192-bits

Table 6: Experimental environment configuration description.

Equipment	Version
Operating system	Windows10
CPU	Intel(R)Core(TM)i5-8300HCPU
Memory	8.00 GB
Word size	64 bits
CPU clock speed	2.30 GHz

8. Implementation and Evaluation

The parameter setting in our scheme is given in Table 5 such that the proposed scheme is secure, and we implemented the scheme under the operating environment indicated in Table 6. We ran the signature generation and signature verification algorithms 1000 times, respectively. The specific time comparison results of our scheme and the schemes of [15, 28, 29] at security level $\lambda = 80$ in the case of different numbers of ring members are shown in Table 7 (let the length of q, p be 160 bits and 512 bits, respectively, in the scheme of [15], and let $n = 256, q = 2^{32}, k = 3, l = 4$ in the scheme of [28], respectively; scheme [29] has the same parameter settings as our scheme). According to Table 7, the signature generation and verification time costs of our

Table 7: Comparison of time costs (ms) at security level $\lambda = 80$.

Scheme	Signature time							
	$N = 1$	$N = 8$	$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
[15]	32.08	40.77	110.26	189.69	348.53	666.23	1301.62	2572.40
[28]	33.76	99.41	198.86	231.01	265.16	298.31	331.46	363.61
[29]	3.62	13.52	78.56	149.52	286.27	510.87	986.84	2001.91
Ours	2.14	7.22	39.91	75.26	144.38	261.02	511.74	1012.70
Scheme	Verification time							
	$N = 1$	$N = 8$	$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
[15]	23.42	32.13	101.63	181.05	339.90	657.60	1292.99	2563.77
[28]	13.20	39.65	79.22	92.41	105.60	118.83	135.68	152.46
[29]	1.23	6.64	49.92	94.72	162.75	316.16	622.56	1172.64
Ours	0.74	4.68	26.56	49.92	94.72	162.72	316.16	622.56

Table 8: Time costs of our scheme (ms).

Parameter	Signature time							
	$N = 1$	$N = 8$	$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
I	2.14	7.22	39.91	75.26	144.38	261.02	511.74	1012.70
II	2.44	8.63	46.46	87.26	164.54	321.82	618.30	1168.86
Parameter	Verification time							
	$N = 1$	$N = 8$	$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
I	0.74	4.68	26.56	49.92	94.72	162.72	316.16	622.56
II	0.84	5.52	29.92	55.36	101.92	197.76	371.36	676.16

scheme are shorter than those of [15, 29]. Compared with the scheme in [28], the proposed scheme has higher signature calculation efficiency when $N \leq 512$. When $N \leq 256$, the verification time cost of the proposed scheme is smaller than that of [28]. However, when the ring members N are large, the signature generation and verification efficiency of our scheme need to be improved compared with [28]. Finally, we use Table 8 to show the specific time costs of the signature generation and verification of our scheme under two different parameter types (I, II) when the number of ring members is different. After calculation, on average, the signature generation time cost of our scheme decreases roughly by 44.951%, and the signature verification time cost also decreases roughly by 33.503% compared with the other three schemes [15, 28, 29]. To show the advantage of our scheme in view of time costs more intuitively, we draw Figures 2 and 3. They, respectively, depict the signature generation and verification time costs of our scheme compared with other schemes under different numbers of ring members at security level $\lambda = 80$. In summary, our scheme achieves relatively higher computational efficiency.

The size of public/private key and signature of the schemes of [15, 27, 28] and our scheme at security level $\lambda = 80$ under different numbers of ring members are listed in Table 9. On the size of public key, the public key size of our scheme is equal to that of scheme [29] but is significantly smaller than that of scheme [28]. In terms of private key size, the scheme in [29] has the biggest private key, and the scheme of [15] has the smallest private key. However, the scheme of [15] is designed based on classical number theory problem (DBDH) and cannot resist quantum attacks. Compared with other lattice-based schemes, the proposed

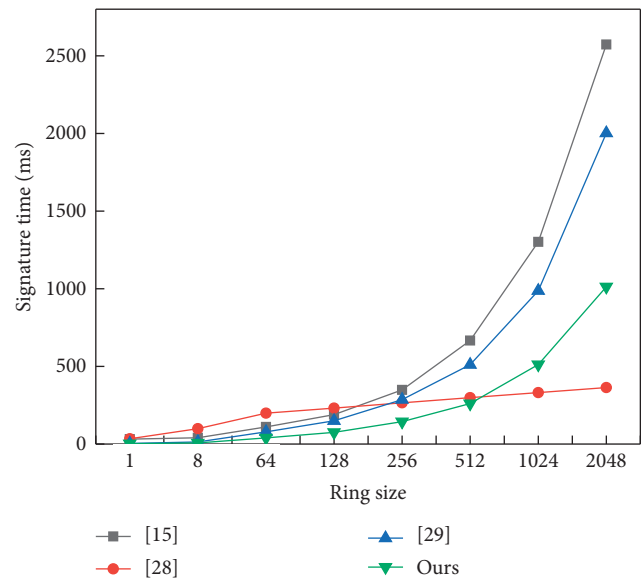

Figure 2: Comparison of time costs at security level $\lambda = 80$. The time cost of signature generation of schemes [15,28,29] is compared with our scheme in the case of different numbers of ring members.

scheme has obvious advantages in key storage overhead. In view of signature size, the signature length of our scheme is short and is significantly better than that of [29]. When $N \geq 64$, the signature length of scheme [28] is shorter than that of our scheme, but note that scheme of [28] is not identity-based, which has the problem of certificate management. After further calculation, the size of signature decreases roughly by 32.078% compared with the scheme of

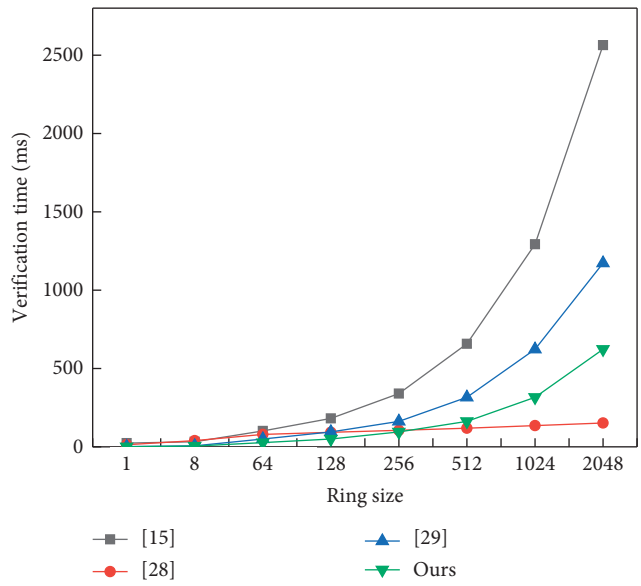

Figure 3: Comparison of time costs at security level $\lambda = 80$. The time cost of signature verification of schemes [15, 28, 29] is compared with our scheme in the case of different numbers of ring members.

Table 9: Comparison of storage overhead (kB) at security level $\lambda = 80$.

Scheme	[15]	[28]	[29]	Ours
Size of public key	0.61	31.56	2.63	2.63
Size of private key	0.61	10.52	23.67	10.52
Signature size for $N = 1$	3.05	34.19	10.52	6.89
Signature size for $N = 8$	7.32	39.45	65.75	38.71
Signature size for $N = 64$	41.48	47.34	507.59	339.27
Signature size for $N = 128$	80.52	49.97	1012.55	675.91
Signature size for $N = 256$	158.60	52.60	2022.47	1349.19
Signature size for $N = 512$	314.76	55.23	4042.31	2695.75

Table 10: Signature size of our scheme under different parameter settings (kB).

Parameter	I	II
Signature size for $N = 1$	6.89	15.69
Signature size for $N = 8$	38.71	88.91
Signature size for $N = 64$	339.27	674.67
Signature size for $N = 128$	675.91	1344.11
Signature size for $N = 256$	1349.19	2682.99
Signature size for $N = 512$	2695.75	5360.75

[29] on average. Finally, we give the signature size of the proposed scheme under different parameter types (I, II) and different number of ring members in Table 10.

9. Conclusions

Linkable ring signature performs a very important role in cryptography. Compared with ordinary ring signatures, it could not only protect the user's identity privacy but also detect whether a user has completed two or more signatures with the same private key by running the linking algorithm. Moreover, lattice-based linkable ring signature can resist the attacks of quantum algorithms. When applied to the blockchain, besides protecting the privacy of both parties in the transaction, it can effectively prevent the emergence of "double-spending" problems. This paper based on the NTRU-SIS assumption constructed an identity-based LRS scheme over NTRU lattice [40]. Performance analysis and experiments show that this scheme has a smaller size of key and signature, thus reducing storage overhead. Since the NTRU lattice is a public key cryptosystem on account of the polynomial ring, the calculation process only involves multiplication in the polynomial ring and modular operations of small integers, which further improves the productivity of signature generation and signature verification of this scheme. This scheme has higher computation performance and lower communication and storage overhead, and it can be applied to more application scenarios than ordinary ring signature.

Appendix

A. Identity-Based LRS: Correctness Requirements

In this part, we prove that this scheme has correctness, unconditional anonymity, unforgeability, and linkability under the random oracle model (ROM).

A.1. Correctness of SigGen

Proof. Suppose signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1\le i\le N}, v, I)$ is valid and generated by a member of the ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$; then, the following equations hold:

$$\begin{aligned}
\sum_{i=1}^{N}\left(\mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h\right) - I * v &= \sum_{i\in\{1,2,\cdots N\}, i\neq k}\left(\mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h\right) + \left(\mathbf{z}_{k,1} + \mathbf{z}_{k,2} * h\right) - I * v,\\
&= \sum_{i\in\{1,2,\ldots,N\}, i\neq k}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right) + \left(s_1 + \mathbf{s}_1^*\right) * v + \mathbf{y}_{k,1} + \left[\left(s_2 + \mathbf{s}_2^*\right) * v + \mathbf{y}_{k,2}\right] * h - I * v,\\
&= \sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right) + \left(s_1 + \mathbf{s}_1^*\right) * v + \left(s_2 + \mathbf{s}_2^*\right) * v * h - I * v,\\
&= \sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right) + \left(s_1 + s_2 * h\right) * v + \left(\mathbf{s}_1^* + \mathbf{s}_2^* * h\right) * v - I * v,\\
&= \sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right) + I * v - I * v,\\
&= \sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right).
\end{aligned} \tag{A.1}$$

From the signing process, we can easily to know that the following equation is true:

$$H_2\left(\sum_{i=1}^{N}\mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h - I * v, R, m, I\right) = H_2\left(\sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h,\ R, m, I\right)\right) = v. \tag{A.2}$$

When $i \neq k$, $\mathbf{z}_{i,1} = \mathbf{y}_{i,1}$, $\mathbf{z}_{i,2} = \mathbf{y}_{i,2}$, where $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow\!\!\!- D_\sigma^n$, according to Lemma 1, we know that $\Pr[\mathbf{z}_i \longleftarrow\!\!\!- D_\sigma^n: \|\mathbf{z}_i\| > 2\sigma\sqrt{n}] < 2^{-n}$, satisfying $\|\mathbf{z}_{i,1}\| \le 2\sigma\sqrt{n}$, $\|\mathbf{z}_{i,2}\| \le 2\sigma\sqrt{n}$ with overwhelming probability. When $i = k$, we have $\mathbf{z}_{i,2} = (s_2 + \mathbf{s}_2^*)v + \mathbf{y}_{i,2}$, $\mathbf{z}_{i,1} = (s_1 + \mathbf{s}_1^*)v + \mathbf{y}_{i,1}$. $\mathbf{z}_{i,1}$ and $\mathbf{z}_{i,2}$ generated by the rejection sampling algorithm, where $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow\!\!\!- D_\sigma^n$, according to Lemma 3 and Theorem 2, are statistically indistinguishable from Gaussian distribution D_σ^n (the statistical distance is $2^{-\omega}(\log n)/M$). Therefore, $\|\mathbf{z}_{i,1}\| \le 2\sigma\sqrt{n}$ and $\|\mathbf{z}_{i,2}\| \le 2\sigma\sqrt{n}$ for $i \in \{1, 2, \ldots, N\}$ are established with overwhelming probability. The proposed identity-based LRS scheme meets verification correctness. □

A.2. Correctness of SigLink

Proof. Suppose the signer uses the same private key $\mathrm{sk}_i = (s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*)$ to sign message $m_1 \in \{0,1\}^*$ and message $m_2 \in \{0,1\}^*$, respectively. There are $I_{(1)} = (s_1 + s_2 * h) + (\mathbf{s}_1^* + \mathbf{s}_2^* * h)$ and $I_2 = (s_1 + s_2 * h) + (\mathbf{s}_1^* + \mathbf{s}_2^* * h)$. $I_{(1)}$ and $I_{(2)}$ are obtained by the same randomly generated polynomial $h \in \mathbf{R}_q$. If a signer calculates signature with the same private key $\mathrm{sk}_i = (s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*)$, $I_{(1)} = I_{(2)}$. The proposed identity-based LRS scheme meets linking correctness. So, our scheme is of correctness. □

B. Security Analysis: Unconditional Anonymity

Proof. The game between a challenger C and an adversary A is used to prove unconditional anonymity. If A is computationally indistinguishable for the two signature distributions, the proposed scheme meets unconditional anonymity. Consider the game indicated below:

(1) Setup phase: C enters a security parameter λ and the number of ring members N and does these steps as follows:
   (a) Sets a set of ring user identities $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$.
   (b) Chooses two collision-resistance hash functions $H_1: \{0,1\}^* \longrightarrow \mathbb{Z}_q^n, H_2: \{0,1\}^* \longrightarrow \{0,1\}^n$ at random.

(c) Uses algorithm $\mathrm{TrapGen}_{\mathrm{NTRU}}(q, n, s)$ to generate a uniform and randomized polynomial $h \in \mathbf{R}_q$ together with a short basis $\mathbf{B} \in \mathbb{Z}_q^{2n\times 2n}$ on lattice $\Lambda_{q,h}$.

(d) Computes the public key $\mathbf{t}_{i,1} = H_1(\mathrm{ID}_i) \in \mathbb{Z}_q^n$ of user ID_i and runs CGS sampling algorithm to generate $(s_1, s_2) = (\mathbf{t}_{i,1}, 0) - \mathrm{CGS}(\mathbf{B}, \sigma, (\mathbf{t}_{i,1}, 0)$; then, $s_1 + s_2 * h = \mathbf{t}_{i,1}$, and it chooses at random $\mathbf{s}_1^*, \mathbf{s}_2^* \longleftarrow\!\!- D_\sigma^n$ and returns the private key $\mathrm{sk}_i = (s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*)$.

(e) Returns the public parameters $\mathrm{PP} = (h, H_1, H_2)$ and the public keys pk_i of ID_i to A and keeps the system master private key $\mathrm{MSK} = \mathbf{B}$ and user private keys sk_i secret, for $i \in \{1, 2, \ldots, N\}$.

(2) Query phase: A is allowed to make adaptive inquiries to above oracles.

(a) Hash query:

(i) H_1 query: A inputs user's identity $\mathrm{ID}_i \in R$ and C returns vector $\mathbf{t}_{i,1}$ to A.

(ii) H_2 query: A inputs a message $m \in \{0,1\}^*$, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, and linkability tag I and randomly chooses $2N$ polynomial vector $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow\!\!- D_\sigma^n$, $i \in \{1, 2, \ldots, N\}$. C randomly chooses an integer v to A.

(b) Corruption query: A inputs user's identity $\mathrm{ID}_i \in R$, and C gives the private key sk_i.

(c) Signing query: A inputs ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and a user's identity $\mathrm{ID}_i \in R$, and C runs $\mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_i)$ algorithm and returns a signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1\le i\le N}, v, I)$ to A.

(3) Challenge phase: A inputs a message $m^* \in \{0,1\}^*$, ring user identity set $R^* = (\mathrm{ID}_1^*, \mathrm{ID}_2^*, \ldots, \mathrm{ID}_N^*)$, and two users' identity $\mathrm{ID}_{i_0}^*, \mathrm{ID}_{i_1}^* \in R^*$, and C selects a random bit $b \in \{0,1\}$, calculates $\mathrm{ID}_{i_b}^*$ corresponding signature private key sk_i^* and then runs Sign algorithm, and returns $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1\le i\le N}, v^*, I^*)$ as the signature of user $\mathrm{ID}_{i_b}^*$ on the message $m^* \in \{0,1\}^*$.

(4) Guess phase: A gives the guess $b^* \in \{0,1\}$ and satisfies $\mathrm{ID}_{i_0}^*, \mathrm{ID}_{i_1}^* \in R^*$ which have not been input to CO and SO at the same time.

Analysis. Next, we analyze the advantage $\mathrm{Adv}_A^{\mathrm{anon}} = |\Pr[b^* = b] - 1/2| = \varepsilon$ of A in winning the game of unconditional anonymity which is negligible. It just needs to explain that the distribution of $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1\le i\le N}, v^*, I^*)$ generated with the sk_i of user $\mathrm{ID}_{i_0}^*$ and $\sigma_{R_\prime}(m\prime) = ((\mathbf{z}_{i,1}', \mathbf{z}_{i,2}')_{1\le i\le N}, v\prime, I\prime)$ generated with the sk_i of user $\mathrm{ID}_{i_1}^*$ by the challenger C is computationally indistinguishable.

According to the signing process, the signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1\le i\le N}, v, I)$ is generated by a randomly selected user ID_i in ring $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$. It constructs a signature based upon the fact that a public key matches multiple secret keys in this scheme. The identity ID_i corresponds to each possible actual signer. There is a private key $\mathrm{sk}_i = (s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*)$ uniquely corresponding to the linkability tag $I \in \mathbf{R}_q$. The signature $\sigma_R(m)$ can be generated by any signer who has a private key sk_i and randomly selected polynomial vectors $(\mathbf{y}_{i,1}, \mathbf{y}_{i,2})_{1\le i\le N}$.

For $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1\le i\le N}, v^*, I^*)$, when $i \neq i_b$, polynomial vectors $\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^* \longleftarrow\!\!- D_\sigma^n$; when $i = i_b$, polynomial vectors $\mathbf{z}_{i,1}^* = (s_1 + \mathbf{s}_1^*)v^* + \mathbf{y}_{i,1}, \mathbf{z}_{i,2}^* = (s_2 + \mathbf{s}_2^*)v^* + \mathbf{y}_{i,2}$, and $\mathbf{z}_{i,1}$ and $\mathbf{z}_{i,2}$ are obtained by using rejection sampling algorithm, where $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow\!\!- D_\sigma^n$. According to Lemma 3 and Theorem 2, $\mathbf{z}_{i_b}^* = (\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)$ is statistically close to $(D_\sigma^n)^2$ (the statistical distance is $(2^{-\omega(\log n)}/M)$). Linkability tag $I^* \in \mathbf{R}_q$ is statistically close to random distribution $\mathbf{R}_q$. So, $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1\le i\le N}, v^*, I^*)$ and $(D_\sigma^n)^{2(N+1)}$ are indistinguishable. Similarly, for $\sigma_{R_\prime}(m') = ((\mathbf{z}_{i,1}', \mathbf{z}_{i,2}')_{1\le i\le N}, v', I')$, when $i \neq i_{1-b}$, polynomial vectors $\mathbf{z}_{i,1}', \mathbf{z}_{i,2}' \longleftarrow\!\!- D_\sigma^n$; when $i = i_{1-b}$, polynomial vectors $\mathbf{z}_{i,1}' = (s_1 + \mathbf{s}_1^*)v\prime + \mathbf{y}_{i,1}$, $\mathbf{z}_{i,2}' = (s_2 + \mathbf{s}_2^*)v' + \mathbf{y}_{i,2}$. $\mathbf{z}_{i,1}'$ and $\mathbf{z}_{i,2}'$ are obtained by using rejection sampling algorithm, where $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow\!\!- D_\sigma^n$. According to Lemma 3 and Theorem 2, $\mathbf{z}_{i_{1-b}}{}' = (\mathbf{z}_{i,1}', \mathbf{z}_{i,2}')$ is statistically close to $(D_\sigma^n)^2$ (the statistical distance is $(2^{-\omega(\log n)}/M)$). Linkability tag $I' \in \mathbf{R}_q$ is statistically close to random distribution $\mathbf{R}_q$. So, the signatures $\sigma_{R_\prime}(m) = ((\mathbf{z}_{i,1}', \mathbf{z}_{i,2}')_{1\le i\le N}, v', I')$ and $(D_\sigma^n)^{2(N+1)}$ are indistinguishable. Therefore, the two signatures $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1\le i\le N}, v^*, I^*)$ and $\sigma_{R_\prime}(m') = ((\mathbf{z}_{i,1}', \mathbf{z}_{i,2}')_1 \le i \le N, v', I')$ have the same discrete Gaussian distribution, and the distribution between two signatures is computationally indistinguishable. The signature can be generated by any user ID_i^* holding the private key and polynomial vectors $(\mathbf{y}_{i,1}, \mathbf{y}_{i,2})_{1\le i\le N}$ can be randomly chosen. Even if A with unbounded computation power can calculate the private key sk_i of the ring member ID_i, since the private key obeys a random distribution, the private key that uniquely matches the linkability tag cannot be calculated. That is, the correct value of $b^* \in \{0,1\}$ cannot be output with a probability better than random guessing. The probability of A giving right guess $b \in \{0,1\}$ can be neglected. So, our scheme has unconditional anonymity. □

C. Security Analysis: Unforgeability

Proof. The game between a challenger C and an adversary A is used to prove unforgeability. Suppose that the signature is successfully forged by A with a non-negligible probability ε. We will show how C uses the forged results of A to find a set of non-zero small-size polynomials $(u, v) \in \mathbf{R}_q^2$ satisfying $u + v * h = 0 \bmod q$ to construct a solution of the SIS problem on NTRU lattice. Hash functions H_1, H_2 are treated as random oracles, and C creates four lists L_1, L_2, L_3, L_4 to store H_1 – oracle queries, H_2 – oracle queries, corruption queries, and signing queries of A. All four lists are initialized to empty.

Now consider the game as indicated below:

(1) Setup phase: To solve the NTRU-SIS problem, C obtains an instance h. Then,inputs security parameter λ, and the number of ring members N, does these steps as following:

(a) Sets a set of ring user identities $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$.
(b) Chooses two collision-resistance hash functions $H_1: \{0,1\}^* \longrightarrow \mathbb{Z}_q^n, H_2: \{0,1\}^* \longrightarrow \{0,1\}^n$ at random.
(c) Calculates the public key $\mathbf{t}_{i,1} = H_1(\mathrm{ID}_i) \in \mathbb{Z}_q^n$ of user ID_i.
(d) Outputs $\mathrm{PP} = (R, h, H_1, H_2)$ as public parameters.

(2) Query phase: A is allowed to make adaptive inquiries to above oracles.

(a) Hash query:
(i) H_1 query: A inputs user's identity $\mathrm{ID}_i \in R$ to query. C checks list L_1; if A has made the same inquiry, it returns the same inquiry result. Otherwise, it returns vector $\mathbf{t}_{i,1} = H_1(\mathrm{ID}_i)$ to A. It adds $(\mathrm{ID}_i, \mathbf{t}_{i,1})$ to the list L_1.
(ii) H_2 query: A inputs a message $m \in \{0,1\}^*$, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, and linkability tag I and randomly chooses $2N$ polynomial vectors $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow D_\sigma^n$, $i \in \{1, 2, \ldots, N\}$ to query. C checks list L_2. If A has made the same inquiry, it returns the same inquiry result. Otherwise, C randomly chooses an integer v to A. It adds $(m, R, I, (\mathbf{y}_{i,1}, \mathbf{y}_{i,2})_{1 \le i \le N}, v)$ to the list L_2.
(b) Registration query: A inputs user's identity ID_i to query; suppose that A can only perform query RO for N^* times at most, where $N^* \ge N$. C selects a subset X_N with N indexes at random. We use $1, 2, \ldots, N$ to define the index of $\mathrm{ID}_i \in R$ (where C does not know the associated private key) and use $N+1, N+2, \ldots, N^*$ to denote the index of $\mathrm{ID}_i \notin R$. When $\mathrm{ID}_i \in R$, it sets vector $\mathbf{t}_{i,1}$ as the public key corresponding to each index in X_N; when $\mathrm{ID}_i \notin R$, C calculates the public key $\mathbf{t}_{i,1}$ by $\mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}_i, \mathrm{MSK})$ algorithm. Upon the ith query, C gives the corresponding public key pk_i. It adds the new tuple $(\mathrm{ID}_i, \mathbf{t}_{i,1})$ to the list L_1.
(c) Corruption query: A inputs user's identity ID_i to query; if $\mathrm{ID}_i \in R$, C halts. Otherwise, C gives the corresponding private key sk_i through the $\mathrm{KeyGen}(\mathrm{PP}, \mathrm{ID}_i, \mathrm{MSK})$ algorithm. It adds $(\mathrm{ID}_i, \mathrm{sk}_i)$ to the list L_3.
(d) Signing query: A inputs a new ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and a user identity ID_i to query. C simulates the following two different situations:
(i) If $\mathrm{ID}_i \notin X_N$, C checks the lists L_2, L_3 and finds the corresponding records $(m, R, I, (\mathbf{y}_{i,1}, \mathbf{y}_{i,2})_{1 \le i \le N}, v)$ and $(\mathrm{ID}_i, \mathrm{sk}_i)$ does these steps (if the lists L_2, L_3 are empty, C obtains the signature based on $\mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_k)$ algorithm):
(1) If $i \ne k$, it sets $\mathbf{z}_{i,1} = \mathbf{y}_{i,1}$, $\mathbf{z}_{i,2} = \mathbf{y}_{i,2}$; if $i = k$, it calculates
$$\mathbf{z}_i = \begin{pmatrix} \mathbf{z}_{i,1} \\ \mathbf{z}_{i,2} \end{pmatrix} = \begin{pmatrix} s_1 + \mathbf{s}_1^* \\ s_2 + \mathbf{s}_2^* \end{pmatrix} * v + \begin{pmatrix} \mathbf{y}_{i,1} \\ \mathbf{y}_{i,2} \end{pmatrix}.$$
(2) By probability $D_{\widetilde{o}(\sqrt{2n}\sigma)}((\mathbf{z}_{i,1}, \mathbf{z}_{i,2}))$ $/MD_{((s_1+s_1^*)v,(s_2+s_2^*)v),\widetilde{o}(\sqrt{2n}\sigma)}((\mathbf{z}_{i,1}, \mathbf{z}_{i,2}))$, it outputs the signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$.
(ii) If $\mathrm{ID}_i \in X_N$, C does these steps:
(1) Randomly chooses polynomial vectors $\mathbf{z}_{i,1}, \mathbf{z}_{i,2} \longleftarrow D_\sigma^n$, $i \in \{1, 2, \ldots, N\}$.
(2) Checks the lists L_2 and finds the corresponding records $(m, R, I, (\mathbf{y}_{i,1}, \mathbf{y}_{i,2})_{1 \le i \le N}, v)$; if H_2 is not queried, perform the H_2 query according to the above steps.
(3) Sets $H_2(\sum_{i=1}^N \mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h - I * v, R, m, I) = v$; if collision occurs, that is, the value v has been assigned to some H_2 query, repeat the above steps.
(4) Outputs the signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$.

(3) Forgery phase: A submits a signature $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1 \le i \le N}, v^*, I^*)$ after the simulation.

Analysis. First, for each different H_2 query, the value v returned by C is randomly selected. It is the same as the randomly distributed value output by the H_2 function in the real life. For the signature query of message $m \in \{0,1\}^*$, polynomial vectors $\mathbf{z}_{i,1}, \mathbf{z}_{i,2} \longleftarrow D_\sigma^n$, $i \in \{1, 2, \ldots, N\}$ in the returned signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$. $\|\mathbf{z}_{i,1}\| \le 2\sigma\sqrt{n}$, $\|\mathbf{z}_{i,2}\| \le 2\sigma\sqrt{n}$ and $H_2(\sum_{i=1}^N \mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h - I * v, R, m, I) = v$. Therefore, $\sigma_R(m)$ is a legal signature.

If the forgery of $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1 \le i \le N}, v^*, I^*)$ is valid, the following will show how C uses the forged results of A to solve NTRU-SIS problem. We mainly analyze from the following two situations:

(i) If v^* appears in the signing query, suppose the output of the query is $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v^*, I)$. Since the signature is a valid signature, it satisfies
$$v^* = H_2\left(\sum_{i=1}^N (\mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h) - I * v^*, R, m, I\right). \quad (C.1)$$
If A successfully forges the signature $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1 \le i \le N}, v^*, I^*)$, we have
$$v^* = H_2\left(\sum_{i=1}^N (\mathbf{z}_{i,1}^* + \mathbf{z}_{i,2}^* * h) - I^* * v^*, R^*, m^*, I^*\right). \quad (C.2)$$
If H_2 function collision occurs, C aborts the game. Otherwise, from (C.1) and (C.2):

$$R^* = R, \quad m^* = m, \quad I^* = I. \tag{C.3}$$

That is,

$$\sum_{i=1}^{N}\left(\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1}\right) + \left(\mathbf{z}_{i,2}^* - \mathbf{z}_{i,2}\right) * h = 0 \bmod q. \tag{C.4}$$

Set $\mathbf{z}_1 = \sum_{i=1}^{N}(\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1}), \mathbf{z}_2 = \sum_{i=1}^{N}(\mathbf{z}_{i,2}^* - \mathbf{z}_{i,2})$, where $[\mathbf{z}_1, \mathbf{z}_2]$ is the answer to the problem of NTRU-SIS.

(ii) If v^* appears in the H_2 query, C finds $(m, R, I, (\mathbf{y}_{i,1}, \mathbf{y}_{i,2})_{1 \le i \le N}, v^*)$ in list L_2, satisfying

$$v^* = H_2\left(\sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right), R, m, I\right). \tag{C.5}$$

If H_2 function collision occurs, C aborts the game. Otherwise, from (C.2) and (C.5):

$$R^* = R, \quad m^* = m, \quad I^* = I, \tag{C.6}$$

$$\sum_{i=1}^{N}\left(\mathbf{z}_{i,1}^* + \mathbf{z}_{i,2}^* * h\right) - I^* * v^* = \sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right). \tag{C.7}$$

C does as follows: if $i \neq k^*$, it sets $\mathbf{z}_{i,1} = \mathbf{y}_{i,1}$, $\mathbf{z}_{i,2} = \mathbf{y}_{i,2}$; if $i = k^*$, it sets $\mathbf{z}_{i,1} = \mathbf{y}_{i,1} + I * v^*$, $\mathbf{z}_{i,2} = \mathbf{y}_{i,2}$. We have

$$\begin{aligned} v^* &= H_2\left(\sum_{i=1}^{N}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right), R, m, I\right), \\ &= H_2\left(\sum_{1 \le i \le N, i \neq k}\left(\mathbf{y}_{i,1} + \mathbf{y}_{i,2} * h\right) + \left(\mathbf{y}_{k,1} + \mathbf{y}_{k,2} * h\right), R, m, I\right), \\ &= H_2\left(\sum_{1 \le i \le N, i \neq k}\left(\mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h\right) + \left(\mathbf{z}_{k,1} + \mathbf{z}_{k,2} * h\right) - I * v^*, R, m, I\right), \\ &= H_2\left(\sum_{1 \le i \le N}\left(\mathbf{z}_{i,1} + \mathbf{z}_{i,2} * h\right) - I * v^*, R, m, I\right). \end{aligned} \tag{C.8}$$

According to (C.2), (C.6), and (C.8), the signature $\sigma_{R^*}(m^*) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v^*, I)$ is valid. That is,

$$\sum_{i=1}^{N}\left(\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1}\right) + \left(\mathbf{z}_{i,2}^* - \mathbf{z}_{i,2}\right) * h = 0 \bmod q. \tag{C.9}$$

If $\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1} = 0$, $\mathbf{z}_{i,2}^* - \mathbf{z}_{i,2} = 0$, C aborts the game.

If $\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1} \neq 0$, $\mathbf{z}_{i,2}^* - \mathbf{z}_{i,2} \neq 0$, let $\mathbf{z}_{i,1}' = \mathbf{z}_{i,1}^* - \mathbf{z}_{i,1}$, $\mathbf{z}_{i,2}' = \mathbf{z}_{i,2}^* - \mathbf{z}_{i,2}$. Also, the following holds:

$$\begin{aligned} \left\|\mathbf{z}_{i,1}'\right\| &= \left\|\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1}\right\| \le \left\|\mathbf{z}_{i,1}^*\right\| + \left\|\mathbf{z}_{i,1}\right\| \le 2\sigma\sqrt{n} + 2\sigma\sqrt{n} = 4\sigma\sqrt{n}, \\ \left\|\mathbf{z}_{i,2}'\right\| &= \left\|\mathbf{z}_{i,2}^* - \mathbf{z}_{i,2}\right\| \le \left\|\mathbf{z}_{i,2}^*\right\| + \left\|\mathbf{z}_{i,2}\right\| \le 2\sigma\sqrt{n} + 2\sigma\sqrt{n} = 4\sigma\sqrt{n}. \end{aligned} \tag{C.10}$$

Set $\mathbf{z}_1 = \sum_{i=1}^{N}\mathbf{z}_{i,1}'$, $\mathbf{z}_2 = \sum_{i=1}^{N}\mathbf{z}_{i,2}'$, where $[\mathbf{z}_1, \mathbf{z}_2]$ is a solution of the NTRU-SIS problem.

Probability Analysis. We assume that A can successfully forge with probability ε and then analyze the probability ε^* that C can successfully find $[\mathbf{z}_1, \mathbf{z}_2]$. C will abandon and abort the game in the following cases, and then the simulation failed.

(1) When H_2 function collision occurs, the probability of signature $\sigma_{R^*}(m^*)$ being verified is $(1/2^n)$.

(2) When $\mathbf{z}_{i,1}^* - \mathbf{z}_{i,1} = 0, \mathbf{z}_{i,2}^* - \mathbf{z}_{i,2} = 0$. This means that the private key sk_i matching signature $\sigma_R(m)$ and the private key sk_i^* corresponding to the forged signature $\sigma_{R^*}(m^*)$ are equal. In the view of A, the signature and the private key are independent of each other when the private key is not known. Therefore, the probability ε_1 that $\mathrm{sk}_i = \mathrm{sk}_i^*$ is negligible.

Hence, we have a probability of higher than $\varepsilon^* \ge \varepsilon - (1/2^n) * 2 - \varepsilon_1 = \varepsilon - (1/2^{n-1})$ to solve the difficult problem of NTRU-SIS. This is in contradiction to the assumption. So, our scheme has unforgeability. □

D. Security Analysis: Linkability

Proof. The game between a challenger C and an adversary A is used to prove the linkability. According to the definition of linkability, assume that adversary A can win the linkability game in Definition 12 with a non-negligible probability ε.

Now consider the game as indicated below:

(1) C uses algorithm $\mathrm{Setup}(1^\lambda, 1^N)$ to obtain the public parameters PP and system master private key MSK and returns the public parameters PP to A.

(2) A is allowed to make adaptive inquiries to above oracles.

(a) Hash query:

(i) H_1 query: A inputs user's identity $\mathrm{ID}_i \in R$, and C returns vector $\mathbf{t}_{i,1}$ to A.

(ii) H_2 query: A inputs a message $m \in \{0,1\}^*$, ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, and linkability tag I and randomly selects $2N$ polynomial vectors $\mathbf{y}_{i,1}, \mathbf{y}_{i,2} \longleftarrow D_\sigma^n$, $i \in \{1,2,\ldots,N\}$, and C randomly chooses an integer v to A.

(b) Registration query: A inputs the identity of user $\mathrm{ID}_i \in R$, and C gives the public key pk_i to A.

(c) Corruption query: A inputs the identity of user $\mathrm{ID}_i \in R$, and C sends the private key sk_i to A.

(d) Signing query: A inputs ring user identity set $R = (\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_N)$, a message $m \in \{0,1\}^*$, and a user's identity $\mathrm{ID}_i \in R$, and C runs the algorithm $\mathrm{Sign}(\mathrm{PP}, R, m, \mathrm{sk}_i)$ and returns a signature $\sigma_R(m) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_{1 \le i \le N}, v, I)$ to A.

(3) A outputs two signatures $\sigma_{R_1}(m_1) = ((\mathbf{z}_{i,1}, \mathbf{z}_{i,2})_1 \le i \le N, v, I_{(1)})$, $\sigma_{R_2}(m_2) = ((\mathbf{z}_{i,1}^*, \mathbf{z}_{i,2}^*)_{1 \le i \le N}, v^*, I_{(2)})$, and the following is satisfied:

(a) All of the public keys in $R_i, i \in \{1,2\}$ are outputs of RO.

(b) A has inquired CO less than two times (that is, A has one private key of users at most).

(c) $\sigma_{R_1}(m_1), \sigma_{R_2}(m_2)$ are not inquired outputs of SO.

Analysis. Suppose A with a non-negligible probability outputs two ring signatures $\sigma_{R_1}(m_1)$ and $\sigma_{R_2}(m_2)$ while holding only one signature private key, and there is "Valid" $\longleftarrow \mathrm{Verify}(\mathrm{PP}, R_i, m_i, \sigma_{R_i}(m_i))$, $i = \{1,2\}$. Since our scheme is unforgeable, signatures $\sigma_{R_1}(m_1)$ and $\sigma_{R_2}(m_2)$ can be verified by the Verify algorithm which returns "Valid" only when these two signatures are generated by A according to the specification honestly. In other words, $I_{(1)} = \mathbf{t}_{k,1} + t_{k,2} = (s_1 + s_2 * h) + (\mathbf{s}_1^* + \mathbf{s}_2^* * h)$ and $I_{(2)} = \mathbf{t}_{k^*,1} + t_{k^*,2} = (s_3 + s_4 * h) + (\mathbf{s}_3^* + \mathbf{s}_4^* * h)$. Since A only holds one private key, $(s_1, s_2, \mathbf{s}_1^*, \mathbf{s}_2^*) = (s_3, s_4, \mathbf{s}_3^*, \mathbf{s}_4^*)$, where the polynomial $h \in \mathbf{R}_q$ is a randomly chosen public parameter. So, we have $I_{(1)} = I_{(2)}$. It means that these two signatures $\sigma_{R_1}(m_1)$ and $\sigma_{R_2}(m_2)$ will return "Link" when verified by the $\mathrm{Link}(\sigma_{R_1}(m_1), \sigma_{R_2}(m_2))$ algorithm. This is in contradiction to the assumption in Definition 12; the advantage $\mathrm{Adv}_A^{\mathrm{link}}$ of A is negligible. So, our scheme has linkability. □

Data Availability

Our results are available on https://github.com/xff-github/NTRU.git.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (grant no. 61802117), Support Plan of Scientific and Technological Innovation Team in Universities of Henan Province (grant no. 20IRTSTHN013), and the Youth Backbone Teacher Support Program of Henan Polytechnic University (grant no. 2018XQG-10).

References

[1] S. Nakamoto, "Bitcoin. A peer-to-peer electronic cash system [EB/OL]," 2017, http://www.bitcoin.org/bitcoin. Pdf..

[2] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, Gold Coast, Australia, November 2001.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pp. 47–53, Springer, Santa Barbara, CA, USA, August 1985.

[4] F. Zhang and K. Kim, "ID-based blind signature and ring signature form pairing," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, vol. 58-4, pp. 305–732, Queenstown, New Zealand, November 2002.

[5] C. Y. Liu and T. C. Wu, "An identity-based ring signature scheme from bilinear pairings," in *Proceedings of the 18th International Conference on Advanced Information Networking and Applications*, p. 182, March 2004.

[6] J. Herranz and G. Sáez, "New identity-based ring signature schemes," in *Proceedings of the Information and Communications Security, 6th International Conference, ICICS 2004*, pp. 27–39, Malaga, Spain, 2004.

[7] S. S. M. Chow, S.-M. Yiu, and L. C. K. Hui, "Efficient identity based ring signature," in *Proceedings of the International Conference on Applied Cryptography and Network Security 2005*, pp. 499–512, Springer, New York, NY, USA, June 2005.

[8] Y. Q. Zhao, Q. Q. Lai, Y. Yu et al., "ID-based ring signature in the standard model," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 46, no. 4, pp. 1019–1024, 2018.

[9] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 325–335, Springer, Sydney, Australia, July 2004.

[10] J. K. Liu, M. H. Au, X. Huang, W. Susilo, J. Zhou, and Y. Yu, "New insight to preserve online survey accuracy and privacy in big data era,"vol. 7-11, pp. 182–199, in *Proceedings of the 19th European Symposium on Research in Computer Security*, vol. 7-11, , Springer, Poland, Europe, September 2014.

[11] P. P. Tsang and V. K. Wei, "Short linkable ring signatures for E-voting, E-cash and attestation," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 48–60, Singapore, 2005 April.

[12] S. Noether, "Ring signature confidential transactions for monero," 2015, https://eprint.iacr.org/2015/1098.pdf.

[13] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Secure ID-based linkable and revocable-iff-linked ring signature with

constant-size construction," *Theoretical Computer Science*, vol. 469, pp. 1–14, 2013.

[14] J. K. Liu, M. H. Man Ho Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Transactions on Knowledge & Data Engineering*, vol. 26, pp. 157–165, 2013.

[15] L. Deng, Y. Jiang, and B. Ning, "Identity-based linkable ring signature scheme," *IEEE Access*, vol. 7, pp. 153969–153976, 2019.

[16] T. EIGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, pp. 469–472, 1985.

[17] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[18] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[19] M. Rückert, "Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles," in *Proceedings of the International Workshop on Post-quantum Cryptography*, pp. 182–200, Springer, Darmstadt, Germany, May 2010.

[20] M.-M. Tian, L.-S. Huang, and W. Yang, "Efficient lattice-based ring signature scheme," *Chinese Journal of Computers*, vol. 35, no. 4, pp. 712–718, 2012.

[21] Z. Liu, Y. Hu, X. Zhang, and F. Li, "Efficient and strongly unforgeable identity-based signature scheme from lattices in the standard model," *Security and Communication Networks*, vol. 6, no. 1, pp. 69–77, 2013.

[22] K. Wang, Y. Mu, and W. Susilo, "Identity-based quotable ring signature," *Information Sciences*, vol. 321, pp. 71–89, 2015.

[23] M. Tian and L. Huang, "Identity-based signatures from lattices: simpler, faster, shorter," *Fundamenta Informaticae*, vol. 145, no. 2, pp. 171–187, 2016.

[24] R. Yang, M. H. Au, J. Lai, Q. Zu, and Z. Yu, "Lattice-based techniques for accountable Anonymity: composition of abstract stems protocols and weak PRF with efficient protocols from LWE," 2017, https://eprint.iacr.org/2017/781.

[25] W. A. Torres, R. Steinfeld, A. Sakzad et al., "Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain(lattice ringct v1.0)," 2018, https://eprint.iacr.org/2018/379.pdf.

[26] M. Ajtai, "Generating hard instances of the short basis problem," in *Proceedings of the International Colloquium on Automata, Languages a-nd Programming*, pp. 1–9, Springer, Prague, Czech Republic, July 1999.

[27] C. Baum, H. Lin, and S. Oechsner, "Towards practical lattice-based one-time linkable ring signature," in *Proceedings of the 20th International Conference on Information and Communications Security*, pp. 303–322, Springer, France, October 2018.

[28] W. Beullens, S. Katsumata, and F. Pintore, "Calamari and falafl: logarithmic (linkable) ring signatures from isogenies and lattices," in *Advances in Cryptology– ASIACRYPT 2020. ASIACRYPT 2020*, S. Moriai and H. Wang, Eds., vol. 12492Cham, Springer, 2020.

[29] X. Lu, M. H. Au, and Z. Zhang, "Raptor: a practical lattice-based (linkable) ring signature," in *Applied Cryptography and Network Security. ACNS 2019*, R. Deng, V. Gauthier-Umaña, M. Ochoa, and M. Yung, Eds., vol. 11464Cham, Springer, 2019.

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

[31] D. Stehlé and R. Steinfeld, "Making NTRUEncrypt and NTRUSign as secure as worst-case problems over ideal lattices," vol. 4, 2013, http://eprint.iacr.org/2013/004.

[32] V. Lyubashevsky and T. Prest, "Quadratic time, linear space algorithms for gram-schmidt orthogonalization and Gaussian sampling in structured Lattices," in *Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques(EuroCrypt 2015)*, pp. 789–815, Sofia, Bulgaria, 2015.

[33] J. Groth and M. Kohlweiss, "One-out-of-many proofs: or how to leak a secret and spend a coin," *Advances in Cryptology–EUROCRYPT 2015*, Springer, Berlin, Germany, pp. 253–280, 2015.

[34] L. Ducas, V. Lyubashevsky, and T. Prest, "Efficient identity-based encryption over NTRU lattices," in *Proceedings of the 20th International Conference on the Theory and Applications of Cyrptology and Information Security (ASIACRYPT 2014)*, pp. 22–41, LNCS 8874, Kaoshiung, China, 2014.

[35] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *Advances in Cryptology–EUROCRYPT 2008. EUROCRYPT 2008*, N. Smart, Ed., vol. 4965, pp. 31–51, Springer, Berlin, Heidelberg, 2008.

[36] Y. Chen and P. Q. Nguyen, "Bkz 2.0: better lattice security estimates," in *Proceedings of the 17th. International Conference on the Theory and Application of Cryptology and Information Security*, D. H. Lee and X. Wang, Eds., pp. 1–20pp. 1–, Seoul, South Korea, December 2011.

[37] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *Lecture Notes in Computer Science*, pp. 41–69, Springer, Berlin, Germany, 2011.

[38] R. del Pino, V. Lyubashevsky, G. Neven, and G. Seiler, "Practical quantum-safevoting from lattices," 2017, https://eprint.iacr.org/2017/1235.pdf.

[39] D. Leo, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals–dilithium: digital signatures from module lattices," 2017, http://eprint.iacr.org/2017/633.

[40] N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte, "NTRUSign: digital signatures using the NTRU Lattice," in *Cryptographers Track at the RSA Conference*, pp. 120–140, Springer, San Francisco, CA, USA, April 2003.