A Blind Ring Signature based on the Short Integer Solution Problem

Huy Quoc Le[1(✉)(0000-0003-4862-6048)], Dung Hoang Duong[2], Willy Susilo[2]

[1] Graduate School of Mathematics, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka-shi, Fukuoka-ken, 819-0395, Japan.
q-le@math.kyushu-u.ac.jp
[2] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia.
{hduong,wsusilo}@uow.edu.au

Abstract. A blind ring signature scheme is a combination of a ring signature and a blind signature, which allows not only any member of a group of signers to sign on a message on behalf of the group without revealing its identity but also the user who possesses the message to blind it before sending to the group to be signed. Blind ring signature schemes are essential components in e-commercial, e-voting etc. In this paper, we propose the first blind ring signature scheme based on lattices. More precisely, our proposed scheme is proven to be secure in random oracle model under the hardness of the short integer solution (SIS) problem.

1 Introduction

Ring signatures were first introduced by Rivest et al. [19] in 2001. In such a scheme, a signer within a group can form a ring consisting of members in the group to sign a message on behalf of this ring, without using the secret keys of those members. A verifier can easily verify that the signature belongs to the ring using the ring public keys, but cannot reveal the identity of the signer, hence ensures the anonymity of the signer. Ring signatures can be used for whistle blowing [19] or anonymous membership authentication for ad hoc groups [5]. They can be used to derive other primitives such as deniable ring authentication [23] or perfect concurrent signatures [24]. Due to flexibility (forming a ring and signing messages without a group leader) and anonymity property of ring signatures, there have been recently found interesting applications of ring signatures in cryptocurrencies [21]. Another important kind of protocol that provides anonymity is blind signatures, first proposed by Chaum [6] for untraceable payments in 1983. Blind signatures allow a person to get a message signed by a signer without revealing any information about the message to the signer, and hence which provide the anonymity of the signed message. It therefore makes blind signatures useful in electronic auctions and electronic voting systems.

In some real-life applications, such as banking, we must make a single e-bank system more scalable by supporting many banks and adding some other

properties like strong anonymity of the signing banks and unlinkability of two different signatures. It's therefore necessary to combine blind and ring signatures into one, called *blind ring signatures*. Clearly, blind ring signatures find applications in various real-life scenarios that are required a combination of ring signatures and blind signatures; for examples, multi authority e-voting and distributed e-cash systems. Some examples of such contexts can be found such as in [9, 12, 26].

With the threat of Shor's quantum algorithms [22], the research community has been moving towards to post-quantum cryptography [4] in which lattice-based cryptography is one of the most promising candidates due to its high asymptotic efficiency and parallelism, as well as security under worst-case intractability assumptions. At ASIACRYPT 2009, Lyubashevksy [14] constructed a lattice-based identification scheme based on ideal lattices, and obtained a signature scheme via Fiat-Shamir's transformation [7]. Lyubashevsky later improved to a new signature scheme [15] whose security is based on the SIS problem. At AfricaCrypt 2013 [1], Aguilar-Melchor et al. proposed the first lattice-based ring signature scheme. Their construction is based on the scheme of Lyubashevsky [14] over ideal lattices. In 2018, Wang et al. [25] proposed a construction of ring signature from an improved scheme of Lyubashevsky [15]. Regarding blind signatrues, the first scheme based on ideal lattices was introduced by Rückert [20] at Asiacrypt 2010. Recently, in 2018, Zhang et al. [27] also gave a new post-quantum construction for blind signature.

In this paper, inspired from the work of Rückert [20] and two aforementioned works on ring signature [25] and blind signature [27], we construct, for the first time, a blind ring signature scheme based on lattices. The scheme is provably secure (i.e., anonymous, blind and one-more unforgeable) in the random oracle model under the hardness of the SIS problem. Our work exploits the rejection sampling technique [15] and the trapdoor technique [8], which are fundamental tools used in lattice-based cryptography.

2 Preliminaries

Notations. For a positive integer l, we write $[l]$ for the set $\{1, 2, \cdots, l\}$. A column vector is denoted by small bold letter, e.g., vector $\mathbf{v}$. A matrix is denoted by bold capital letter, e.g., $\mathbf{A}$. Sometimes we write a_i, the i-th component of a vector $\mathbf{a} = (a_1, \cdots, a_n)$, by $\mathbf{a}[i]$. The notation $\mathbf{A}[i]$ is also used to stand for the i-th column of a matrix $\mathbf{A}$. The Gram-Schmidt orthogonal matrix of a matrix $\mathbf{A}$ will be written as $\tilde{\mathbf{A}}$. By notation "$x := a$" we mean that the variable x is assigned the value a or x is defined as a. We write $a \leftarrow_{\$} A$ to say that a is sampled uniformly at random from the discrete set A; while if $\mathcal{D}$ is a probability distribution, then $a \leftarrow \mathcal{D}$ means that a is sampled according to $\mathcal{D}$. In case $\mathcal{A}$ is an algorithm, we write $a \leftarrow \mathcal{A}$ to say that a is an output of $\mathcal{A}$.

A *lattice* is a set of all integral combinations of given linearly independent vectors. Formally, given a matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m] \in \mathbb{R}^{n \times m}$ such that $\mathbf{a}_i$'s are linearly independent, a lattice of *basis* $\mathbf{A}$ is the set $\mathcal{L}(\mathbf{A}) := \{\sum_{i \in [m]} \mathbf{a}_i z_i : z_i \in$

$\mathbb{Z}\}$. For such a lattice, we call n the *dimension* of $\mathcal{L}(\mathbf{A})$. Take for example, for a random matrix $\mathbf{A} \leftarrow_{\$} \mathbb{Z}^{n\times m}$, the following are also lattices, called *q-ary lattices*:

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{v} = \mathbf{A}^\top \mathbf{z} \pmod q \text{ for some } \mathbf{z} \in \mathbb{Z}^n\},$$

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{0} \pmod q, \text{ where } \mathbf{A} \leftarrow_{\$} \mathbb{Z}^{n\times m}\}. \tag{1}$$

The *first minimum* of a lattice $\mathcal{L}$ is defined as $\lambda_1(\mathcal{L}) := \min_{\mathbf{v}\in\mathcal{L}\setminus\{\mathbf{0}\}} \|\mathbf{v}\|$. The *i-th minimum* of a lattice $\mathcal{L}$ of dimension n is denoted by and defined as $\lambda_i(\mathcal{L}) := \min\{r : \dim(\text{span}(\mathcal{L} \cap \mathcal{B}_n(0, r))) \geq i\}$, where $\mathcal{B}_n(0, r) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq r\}$. The γ-SIVP problem is, given a basis $\mathbf{A}$ of a lattice $\mathcal{L}(\mathbf{A})$, to search for a set of n linearly independent lattice vectors $\mathbf{S} \subset \mathcal{L}(\mathbf{A})$ such that $\|\mathbf{S}\| \leq \gamma\lambda_n(\mathbf{A})$.

The security of our blind ring signature scheme will be based on the average-case assumption of the short integer solution (SIS) problem.

Definition 1 (SIS Problem). *Given a random matrix $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, a vector $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^n$ and a positive real number β, the inhomogeneous small integer problem $\mathsf{ISIS}_{q,n,m,\beta}$ is to find a vector $\mathbf{z} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\mathbf{Az} = \mathbf{u}$ (mod q) and $\|\mathbf{z}\| \leq \beta$. In the case $\mathbf{u} = \mathbf{0}$, we have the homogeneous small integer problem, named $\mathsf{SIS}_{q,n,m,\beta}$.*

One can prove that the hardness of SIS and ISIS are essentially equivalent for typical parameters [18, Chapter 4]. The $\mathsf{SIS}_{q,n,m,\beta}$ problem can be seen as an average-case short vector problem on the q-ary lattice $\Lambda_q^\perp(\mathbf{A})$ defined as in Equation (1) which requires to find a sufficiently short nonzero vector in $\Lambda_q^\perp(\mathbf{A})$. The SIS problem was first introduced in by Ajtai in his seminal work [2]. He proved that solving the SIS problem can be reduced to solving certain worst-case problems in lattices. Then Miciancio and Regev [17] gave a more tighten reduction saying that for large enough q, solving SIS as hard as solving $\tilde{O}(\beta\sqrt{n})$-SIVP problem in all lattices in dimension n.

Definition 2 (Discrete Gaussian Distribution, Definition 4.2 of [15]). *The discrete Gaussian distribution over $\mathbb{Z}^m$ centered at some $\mathbf{v} \in \mathbb{Z}^m$ with standard deviation σ is defined as $\mathcal{D}^m_{\mathbf{v},\sigma}(\mathbf{x}) := \rho^m_{\mathbf{v},\sigma}(\mathbf{x})/\rho^m_{\mathbf{v},\sigma}(\mathbb{Z}^m)$, where $\rho^m_{\mathbf{v},\sigma}(\mathbf{x}) := \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m e^{\frac{-\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}}$ and $\rho^m_{\mathbf{v},\sigma}(\mathbb{Z}^m) := \sum_{\mathbf{x}\in\mathbb{Z}^m} \rho^m_{\mathbf{v},\sigma}(\mathbf{x})$.*

Some basic facts relating to the discrete Gaussian distribution are summarized in the following lemmas:

Lemma 1 (Lemma 4.3 in [15]).

(i) For any $k > 0$, $\Pr[|z| > k\sigma, z \leftarrow \mathcal{D}^1_\sigma] \leq 2e^{\frac{-k^2}{2}}$,

(ii) For any $\mathbf{v} \in \mathbb{Z}^m$, and any $\sigma, r > 0$, $\Pr[|\langle \mathbf{x}, \mathbf{v}\rangle| > r : \mathbf{x} \leftarrow \mathcal{D}^m_\sigma] \leq 2e^{-\frac{r^2}{2\|\mathbf{v}\|^2\sigma^2}}$.

Remark 1. In Lemma 1(i), if $k = 12$ then $|z| > 12\sigma$ with probability at most 2^{-100}. Similarly, in Lemma 1(ii), if we choose $r = 12\|\mathbf{v}\|\sigma$ then $|\langle \mathbf{x}, \mathbf{v}\rangle| \geq 12\|\mathbf{v}\|\sigma$ with probability at most 2^{-100}.

Lemma 2 (Lemma 4.4 in [15]). *For any $\eta > 0$, we have* $\Pr[\|\mathbf{z}\| > \eta\sigma\sqrt{m}, \mathbf{z} \leftarrow \mathcal{D}_\sigma^m] \leq \eta^m e^{\frac{m}{2}(1-\eta^2)}$.

Remark 2. In Lemma 2, the function $\eta^m e^{\frac{m}{2}(1-\eta^2)}$ is decreasing either in m (if η fixed) or in η (if m fixed). See Table 1 for example. Clearly we need $\eta > 1$ as small as possible. Hence, with typical large enough m, one usually chooses $\eta \in [1.1, 1.3]$.

	$m = 50$	$m = 100$	$m = 200$
$\eta = 1.1$	0.61601	0.37947	0.14399
$\eta = 1.3$	0.01605	0.00026	0.00000007
$\eta = 3$	9.7×10^{-64}	9.9×10^{-127}	9.7×10^{-253}

Table 1: Some specific values for $\eta^m e^{\frac{m}{2}(1-\eta^2)}$

Lemma 3 (Lemma 4.5 in [15]). *For any $\mathbf{v} \in \mathbb{Z}^m$, if $\sigma = \alpha\|\mathbf{v}\|$ where $\alpha > 0$, we have* $\Pr\left[\mathcal{D}_\sigma^m(\mathbf{x})/\mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{x}) \leq e^{12/\alpha+1/(2\alpha^2)} : \mathbf{x} \leftarrow \mathcal{D}_\sigma^m\right] \geq 1 - 2^{-100}$.

Remark 3. In Lemma 3, if we choose, $\alpha = 12$, i.e., $\sigma = 12\|\mathbf{v}\|$ then $\mathcal{D}_\sigma^m(\mathbf{x})/\mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{x}) \leq e^{1+1/288}$ with probability at least $1 - 2^{-100}$.

Definition 3 (Statistical Distance, Definition 8.5 in [16]). *Let X and X' be two random variables over a countable set S. We define the statistical distance between X and X' by $\Delta(X, X') := \frac{1}{2}\sum_{x \in S} |\Pr[X = x] - \Pr[X' = x]|$.*

Lemma 4 (Triangular Inequality). *Let X_1, X_2 and X_3 be three random variables over a countable set S. We have $\Delta(X_1, X_3) \leq \Delta(X_1, X_2) + \Delta(X_2, X_3)$.*

Lemma 5 (Rejection Sampling, Theorem 4.6 in [15]). *Given a subset $V = \{\mathbf{v} \in \mathbb{Z}^m : \|\mathbf{v}\| \leq T\}$ and a real number $\sigma = \omega(T\log\sqrt{m})$. Define on V a probability distribution $h : V \to \mathbb{R}$. Then there exists a universal upper bound $M = O(1)$ such that the outputs of the following two algorithms $\mathcal{A}$ and $\mathcal{B}$ have a negligible statistical distance of $\Delta(\mathcal{A}, \mathcal{B}) := 2^{-\omega(\log m)}/M$:*

1. *($\mathcal{A}$): $\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow \mathcal{D}_{\mathbf{v},\sigma}^m$, output $(\mathbf{z}, \mathbf{v})$ with probability* $\min(\frac{\mathcal{D}_\sigma^m(\mathbf{z},)}{M\mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{z})}, 1)$.
2. *($\mathcal{B}$): $\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow \mathcal{D}_\sigma^m$, output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$.*

Moreover, the probability that $\mathcal{A}$ outputs something is at least $(1-2^{-\omega(\log m)})/M$. Particularly, if $\sigma = \alpha T$ for any $\alpha > 0$ then $M = e^{12/\alpha+1/(2\alpha^2)}$, $\Delta(\mathcal{A}, \mathcal{B}) = 2^{-100}/M$, and the probability that $\mathcal{A}$ outputs something is at least $(1-2^{-100})/M$.

In order to construct the blind ring signature, we exploit the **trapdoor technique** proposed in [8, Subsection 5.3] to generate necessary keys.

$\underline{\mathsf{TrapGen}(1^n)}$. The algorithm on input the security parameter n, chooses a prime $q = poly(n)$ and an integer $m > 5n \log q$ to output a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{B_A} \in \mathbb{Z}_q^{m \times m}$ with $\|\tilde{\mathbf{B}}_\mathbf{A}\| \leq K := m^{1+\epsilon}$ for any $\epsilon > 0$, where *the distribution of* $\mathbf{A}$ *is statistically close to the uniform over* $\mathbb{Z}_q^{n \times m}$ and the matrix $\mathbf{B_A}$ is a good basis of the lattice $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{Av} = \mathbf{0} \pmod q\}$.

We generalize the trapdoor inversion algorithm SampleISIS in [8, Subsection 5.3] to have the following algorithm:

$\underline{\mathsf{SampleKey}(\mathbf{A}, \mathbf{B_A}, \sigma, \mathbf{T})}$. The algorithm takes as input $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{B_A} \in \mathbb{Z}_q^{m \times m}$ outputted by $\mathsf{TrapGen}(1^n)$, a real number $\sigma \geq K \cdot \omega(\sqrt{\log n})$ and matrix $\mathbf{T} \in \mathbb{Z}_q^{n \times k}$, and returns a random (column) matrix $\mathbf{S} \in \mathbb{Z}^{m \times k}$ such that the j-th column $\mathbf{S}[j] \in D = \{\mathbf{s} \in \mathbb{Z}^m : \|\mathbf{s}\| \leq \sigma\sqrt{m}\}$ for all $j \in [k]$ and that $\mathbf{AS} = \mathbf{T} \pmod q$ with overwhelming probability. The distribution of $\mathbf{S}[j]$ for all $j \in [k]$ is $\mathcal{D}_{\mathbb{Z},\sigma}^m$ statistically close to the uniform distribution over D.

In this work, we also exploit the **commitment function** com maps a pair of two strings $(\mu,\mathbf{t}) \in \{0,1\}^* \times \{0,1\}^n$ (called *committed string*) to a *commitment string* $C := \mathsf{com}(\mu, \mathbf{t}) \in \{0,1\}^n$ to hide the value of the message μ. For security goal, we need com to have two properties: *statistically hiding* and *computationally binding*. The first property ensures that any computationally unbounded algorithm is not able to statistically distinguish two commitment strings C and C' obtained from two distinct committed pairs $(\mu, \mathbf{t}) \neq (\mu', \mathbf{t}')$. The second property says that given a commitment string C obtained from the committed pair of strings $(\mu, \mathbf{t})$ (i.e., $C := \mathsf{com}(\mu, \mathbf{t})$), no polynomia-time algorithm can find another pair $(\mu', \mathbf{t}')$ with $\mu' \neq \mu$ such that $C = \mathsf{com}(\mu', \mathbf{t}')$. See [11, 13, 20] for more details.

3 Blind Ring Signature Schemes

A blind ring signature consists of four algorithms called Setup, KeyGen, Sign and Verify.

- $\mathsf{Setup}(1^\lambda)$ is a probabilistic polynomial-time algorithm which takes as input the security parameter λ and outputs a set of public parameters $\mathcal{P}$.
- $\mathsf{KeyGen}(\mathcal{P})$ is a probabilistic polynomial-time algorithm which takes as input the set of public parameters $\mathcal{P}$ to output a pair of public key (verification key) and secret key (signing key) $(\mathsf{pk}, \mathsf{sk})$ corresponding to a signer of the ring $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$. We denote the set of public keys of the ring R by PK.
- $\mathsf{Sign}(\mathcal{P}, \mathsf{sk}_j, \mu, PK)$ is an interactive polynomial-time protocol of two parties: one is a user and another is a ring of signers $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$. The user, say $\mathcal{U}(\mathcal{P}, PK, \mu)$, chooses a message μ that is blinded as μ^* before sending μ^* to the ring R to be signed. The ring R, in turn, will choose a member, say $\mathcal{S}_j$, who possess the secret key sk_j, written $\mathcal{S}_j(\mathcal{P}, \mathsf{sk}_j)$, as the real signer interacting with the user. Finally, the signer obtains the *blinded signature* Σ^* on μ^* and outputs his *view*, denoted $\mathcal{V}$, (it may that $\mathcal{V} \neq \Sigma^*$), while the user will output the *real* (or *final*) *signature* Σ on the original message μ by

un-blinding Σ^*. The user may get an invalid signature denoted by a failure symbol $\perp$.
- $\mathsf{Verify}(\mathcal{P}, \mu, \Sigma, PK)$ is a deterministic polynomial-time algorithm which takes as input the set of common parameters $\mathcal{P}$, the set of public keys PK, the message μ and the signature Σ on μ, then outputs 1 if the signature is valid and 0 otherwise.

A blind ring signature scheme must have the following properties: *Correctness*, *Anonymity*, *Blindness* and *One-more Unforgeability*. We will make these properties clearer below.

Correctness. *Correctness* requires that the verifier always outputs 1 if it receives a valid signature. Formally, it must hold that

$$\Pr[\mathsf{Verify}(\mathcal{P}, \mu, \Sigma, PK) = 1 : \Sigma \leftarrow \mathsf{Sign}(\mathcal{P}, \mathsf{sk}_j, \mu, PK), \Sigma \neq \perp] = 1.$$

A relaxation for the correctness is that if $\Sigma \leftarrow \mathsf{Sign}(\mathcal{P}, \mathsf{sk}_j, \mu, PK), \Sigma \neq \perp$ then $\mathsf{Verify}(\mathcal{P}, \mu, \Sigma, PK) = 1$ with overwhelming probability. (In our case the probability will be at least $1 - 2^{-100}$.)

Anonymity. The anonymity property ensures that a user is impossible to know which member of the ring was the true signer engaging in the blind ring signature protocol. The definition of the anonymity property is given in the game below. In this game, the attacker acts as a malicious user.

1. **Setup.** The adversary $\mathcal{A}$ outputs the set of common parameters $\mathcal{P}$, the ring of signers $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$, its public keys PK, two distinct indexes j, $i_1 \in [l]$, two secret keys $\mathsf{sk}_j, \mathsf{sk}_{i_1}$ and a message μ. They are sent to the challenger $\mathcal{C}$.
2. **Challenge.** The challenger $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$, then runs Sign on the input $(\mathcal{P}, \mathsf{sk}_{i_b}, \mu, PK)$ to get a blinded signature $\Sigma^*_{i_b}$ on μ. The blinded signature $\Sigma^*_{i_b}$ will be given to the adversary $\mathcal{A}$.
3. **Output.** The adversary outputs a bit b' as a guess of b. He wins the game if $b' = b$.

We say that the blind ring signature achieves anonymity if any adversary $\mathcal{A}$ succeeds in guessing b with probability negligibly close to 1/2. In other words, the advantage of $\mathcal{A}$ in distinguishing Σ_j and Σ_{i_1} is negligible.

Blindness. Blindness is a fundamental property of a blind ring signature saying that all members in the ring do not learn any information about the message received from the user that they are having to sign. The property can be modelled as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the adversary $\mathcal{A}$ plays the role of a dishonest ring of signers R who tries to differentiate two given messages to know which one is being signed.

1. **Setup.** The adversary $\mathcal{A}$ chooses a security parameter λ and chooses a universal set of signers to generate the ring $R^* = \{\mathcal{S}_1, \cdots, \mathcal{S}_L\}$. Then it calls

Algorithm 1 BRS.Setup(1^n)

Input: Security parameter n.
Output: The set of public parameters $\mathcal{P}$=(n, m, q, k, κ, σ, H, σ_1, σ_2, σ_3, M_1, M_2, M_3, $\mathbf{T}$, η)
1: Generate parameters as in Table 2
2: An one-way and collision-resistant hash function $H : \{0,1\}^* \to \{\mathbf{c} \in \{-1,0,1\}^k : \|\mathbf{c}\|_1 \leq \kappa\}$
3: A commonly-used matrix $\mathbf{T} \leftarrow_{\$} {\mathbb{Z}_q}^{m \times k}$
4: Output all as the set $\mathcal{P}$

Algorithm 2 BRS.KeyGen ($\mathcal{P}$)

Input: $\mathcal{P}$=(n, m, q, k, κ, σ, H, σ_1, σ_2, σ_3, M_1, M_2, M_3, $\mathbf{T}$, η)
Output: A key pair $(\mathbf{A}, \mathbf{S})$
1: Run $\mathsf{TrapGen}(1^n)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{B}_A \in \mathbb{Z}_q^{n \times m}$ is a trapdoor of $\mathbf{A}$; /* *the distribution of* $\mathbf{A}$ *is statistically close to the uniform over* $\mathbb{Z}_q^{n \times m}$ */
2: $\mathbf{S} \leftarrow \mathsf{SampleKey}(\mathbf{A}, \mathbf{B}_A, \sigma, \mathbf{T})$, i.e., $\mathbf{AS} = \mathbf{T} \pmod q$ where $\mathbf{S} \in D_d := \{-d, \cdots, 0, \cdots, d\}^{m \times k}$, $d = \sigma\sqrt{m}$; /* *the distribution of* $\mathbf{S}$ *is* $\mathcal{D}_{\mathbb{Z},\sigma}^{m \times k}$ *statistically close to the uniform over* D_d */
3: **return** Public key $\mathbf{A}$ and secret key $\mathbf{S}$

$\mathsf{Setup}(1^\lambda)$ to get the set of public parameters $\mathcal{P}$ according to the security parameter λ and $\mathsf{KeyGen}(\mathcal{P})$ to output the key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i \in [L]}$ for each signer $\mathcal{S}_i$, $i \in [L]$. The adversary $\mathcal{A}$ knows $\mathcal{P}$ and $(\mathsf{pk}_i, \mathsf{sk}_i)_{i \in [L]}$.

2. **Challenge.** The adversary $\mathcal{A}$ chooses a subring $R \subset R^*$, and its corresponding public keys PK, and two messages $\mu_0 \neq \mu_1$, then he sends them to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ will flip a coin $b \in \{0,1\}$ and sets up a blind ring signature protocol taking μ_b and the ring R as input. The adversary $\mathcal{A}$ chooses a signer $\mathcal{S}_j$ in the ring R to sign the hidden form of μ_b and acts as the signer in the protocol. Eventually, $\mathcal{A}$ gets the view $\mathcal{V}_b$ and also the "unblinded" signature $\Sigma_b \neq \perp$. If $\Sigma_b = \perp$, the game is restarted.
3. **Guess.** The adversary $\mathcal{A}$ outputs one value $b' \in \{0,1\}$. The adversary wins the game if $b' = b$.

We say that a ring signature scheme is *blind* if for any adversary $\mathcal{A}$ the success probability in the game is only negligibly larger than 1/2.

One-more Unforgeability. The one-more unforgeability property guarantees that from at most q_S real interactions of the blind ring signature protocol, the user has no capacity of producing $q_S + 1$ valid and different ring signatures. The property is defined by the game below. In this game, the forger will act the behaviour of a malicious user.

1. **Setup.** The forger $\mathcal{F}$ chooses a security parameter λ and chooses a universal set of signers to generate the ring $R^* = \{\mathcal{S}_1, \cdots, \mathcal{S}_L\}$. The challenger $\mathcal{C}$ calls $\mathsf{Setup}(1^\lambda)$ to get the set of public parameters $\mathcal{P}$ and $\mathsf{KeyGen}(\mathcal{P})$ to output

the key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)_{i\in[L]}$ for each signer $\mathcal{S}_i$, $i \in [L]$. Then $\mathcal{C}$ sends to the forger $\mathcal{F}$ the set $\mathcal{P}$ and the set of public keys $\{\mathsf{pk}_i\}_{i\in[L]}$. The set $\{\mathsf{sk}_i\}_{i\in[L]}$ is kept secret.

2. **Queries.** The forger $\mathcal{F}$ adaptively makes queries to the challenger:
   - q_H hash queries to the random oracle which models the hash function H in the real protocol. For each hash query from the adversary, the challenger has to reply with a consistently random value.
   - q_S blind signing queries, each is of the form (μ_i, R_i) where $R_i \subset R^*$. For each signing query, the challenger must answer with a valid blind ring signature.
3. **Output.** The forger $\mathcal{F}$ outputs $q_S + 1$ tuples $\{(\mu_i, R_i, \Sigma_i)\}_{i\in[q_S+1]}, R_i \subset R^*$. He wins the game if $\{\Sigma_i\}_{i\in[q_S+1]}$ are all valid and $(\mu_i, R_i) \neq (\mu_j, R_j)$ for all $i, j \in [q_S + 1]$ and $i \neq j$.

We say that a blind ring signature scheme is *one-more unforgeable* if in the game, $\Pr[\mathcal{F}$ wins$]$ is negligible.

Remark 4. For simplicity, in the proof for the one-more unforgeability property of our proposed scheme, we assume that $R_i = R^*$ for all $i \in [q_S + 1]$, that is, the forger does not want to change the ring of signers at all.

4 Our Blind Ring Signature Scheme

We will present our blind ring signature (named BRS) scheme. The security of BRS bases on the average-case assumption of the SIS problem. The scheme follows the 4-move framework for blind ring signature as reviewed in Section 3. It consists of four algorithms (see Algorithms 1-3 and Figure 1) described as follows:

- We suppose that n is the security parameter. $\mathsf{BRS.Setup}(1^n)$ is called to output a common set of parameters $\mathcal{P}$ (see Algorithm 1). We will mention the role of these parameters and how to set them in Subsection 6.
- Given a matrix $\mathbf{T} \leftarrow_{\$} {\mathbb{Z}_q}^{n\times k}$, to generate public key $\mathbf{A}_i \leftarrow_{\$} {\mathbb{Z}_q}^{n\times m}$ and secret key $\mathbf{S}_i \in {\mathbb{Z}_q}^{m\times k}$ for each signer $\mathcal{S}_i$ in a ring $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$ of l members such that $\mathbf{T} = \mathbf{A}_i\mathbf{S}_i$, we run l times $\mathsf{BRS.KeyGen}$ (see Algorithm 2) which exploits the preimage sample functions (trapdoor functions) mentioned in Subsection 2. The secret key $\mathbf{S}_i$ follows a discrete Gaussian distribution $\mathcal{D}_\sigma^{m\times k}$ and its security is guaranteed by the hardness assumption of the ISIS problem.
- The signing algorithm ($\mathsf{BRS.Sign}$)(see Figure 1) is an interactive protocol between a user $\mathcal{U}$ and a ring $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$. The user $\mathcal{U}$ knows the set of public keys PK and he wants the ring R to sign the message μ. Here we describe the protocol in the case that the ring secretly delegates some signer $\mathcal{S}_j \in R$ to interact with the user. We relatively split the signing interaction into five main phases:

The member $\mathcal{S}_j(\mathcal{P}, \mathbf{S}_j)$ is the signer		**The user $\mathcal{U}(\mathcal{P}, PK, \mu)$**
Phase 1: **for** $i \in [l]$: $\mathbf{s}_i \leftarrow_{\$} \mathcal{D}^m_{\sigma_2}$ $\mathbf{x} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{s}_i \pmod q$		
	$\mathbf{x}$ $\longrightarrow$	**Phase 2:** **for** $i \in [l]$: $\mathbf{a}_i \leftarrow \mathcal{D}^m_{\sigma_3}$ $\mathbf{b} \leftarrow \mathcal{D}^k_{\sigma_1}$, $\mathbf{t} \leftarrow_{\$} \{0,1\}^n$, $\mathbf{w} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{a}_i$, $C := \mathsf{com}(\mu, \mathbf{t})$ $\mathbf{u} = \mathbf{x} + \mathbf{w} + \mathbf{Tb} \pmod q$ $\mathbf{c} = H(\mathbf{u}, C, PK), \mathbf{e} = \mathbf{c} + \mathbf{b}$ Outputs $\mathbf{e}$ with probability $\min\left\{\frac{\mathcal{D}^k_{\sigma_1}(\mathbf{e})}{M_1\cdot\mathcal{D}^k_{\mathbf{c},\sigma_1}(\mathbf{e})}, 1\right\}$
Phase 3: **for** $i \in [l] \setminus \{j\}$: $\mathbf{y}_i = \mathbf{s}_i$ **for** j: $\mathbf{y}_j = \mathbf{s}_j + \mathbf{S}_j\mathbf{e}$ Output $\mathbf{y}_j$ with probability $\min\left\{\frac{\mathcal{D}^m_{\sigma_2}(\mathbf{y}_j)}{M_2\cdot\mathcal{D}^m_{\mathbf{S}_j\mathbf{e},\sigma_2}(\mathbf{y}_j)}, 1\right\}$	$\longleftarrow$	$\mathbf{e}$
	$\Sigma^* = \{\mathbf{y}_i\}_{i\in[l]}$ $\longrightarrow$	**Phase 4:** **for** $i \in [l]$: $\mathbf{z}_i = \mathbf{y}_i + \mathbf{a}_i$ accepts $\mathbf{z}_i$ with probability $\min\left\{\frac{\mathcal{D}^m_{\sigma_3}(\mathbf{z}_i)}{M_3\cdot\mathcal{D}^m_{\mathbf{y}_i,\sigma_3}(\mathbf{z}_i)}, 1\right\}$ That is, **if** $(\exists j$ s.t. $\|\mathbf{z}_j\| > \eta\sigma_3\sqrt{m})$: $\mathsf{result} := ((\mathbf{a}_i)_{i\in[l]}, \mathbf{b}, \mathbf{c}, C)$ **else**: $\mathsf{result} := \mathsf{accept}$ **Output:** $(\mu, \Sigma = ((\mathbf{z}_i)_{i\in[l]}, \mathbf{c}, \mathbf{t}))$ or $\perp$ when $\mathsf{result} \neq \mathsf{accept}$
Phase 5: **if** $(\mathsf{result} \neq \mathsf{accept})$: Parse $\mathsf{result} = ((\mathbf{a}_i)_{i\in[l]}, \mathbf{b}, \mathbf{c}, C)$ $\mathbf{w} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{a}_i$, $\mathbf{v} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{y}_i$ $\mathbf{u} = \mathbf{x} + \mathbf{w} + \mathbf{Tb} \pmod q$ $\mathbf{u}' = \mathbf{w} + \mathbf{v} - \mathbf{Tc} \pmod q$ **if** ($\mathbf{e} - \mathbf{b} = \mathbf{c} = H(\mathbf{u}, C, PK)$ and $\mathbf{c} = H(\mathbf{u}', C, PK)$ and $\exists j$ s.t. $\|\mathbf{y}_j + \mathbf{a}_j\| > \eta\sigma_3\sqrt{m})$: Restart the protocol **Output:** the view $\mathcal{V} = (\mathbf{x}, \mathbf{e}, (\mathbf{s}_i, \mathbf{y}_i)_{i\in[l]})$	$\longleftarrow$	result

Fig. 1: The signing protocol $\mathsf{BRS.Sign}(\mathcal{P}, \mathbf{S}_j, \mu, PK)$, $j \in [l]$, $PK = \{\mathbf{A}_i\}_{i\in[l]}$

- Phase 1: The signer samples randomly a list $\{\mathbf{s}_i\}_{i\in[l]}$ according to the distribution $\mathcal{D}^m_{\sigma_2}$ to compute and then sends the *commitment* $\mathbf{x} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{s}_i$ to the user.
- Phase 2: The user chooses blind factors $\mathbf{a}_i \leftarrow \mathcal{D}^m_{\sigma_3}$ for all $i \in [l]$ and $\mathbf{b} \leftarrow \mathcal{D}^k_{\sigma_1}$. He also chooses a random binary vector $\mathbf{t} \leftarrow_\$ \{0,1\}^n$ then uses the commitment function com to compute the commitment string $C := \mathsf{com}(\mu, \mathbf{t}) \in \{0,1\}^n$. Afterward, he computes $\mathbf{u} = \mathbf{x} + \sum_{i\in[l]} \mathbf{A}_i\mathbf{a}_i + \mathbf{Tb}$ then hash it with C using the hash function H where $H : \{0,1\}^* \to D_c := \{\mathbf{c} \in \{-1,0,1\}^k : \|\mathbf{c}\| \leq \kappa\}$ to get the *challenge* $\mathbf{c}$. To blind the message, the user uses the rejection sampling technique to get the *blinded challenge* $\mathbf{e}$. Finally, the user sends $\mathbf{e}$ to the ring.
- Phase 3: This is the signing phase in which the signer $\mathcal{S}_j$ considers $\mathbf{s}_i$'s sampled in Phase 1 as the partial signatures of other members in the ring on the message μ, while he uses his secret key $\mathbf{S}_j$ to compute his himself partial signature $\mathbf{y}_j = \mathbf{s}_j + \mathbf{S}_j\mathbf{e}$ on μ. In order to make sure that no information of his secret key $\mathbf{S}_j$ is leaked, the signer also exploits the rejection sampling such that $\mathbf{y}_j$ follows the same distribution $\mathcal{D}^m_{\sigma_2}$ as $\mathbf{s}_j$. Finally, he sends the *blinded signature* $\{\mathbf{y}_i\}_{i\in[l]}$ to the user.
- Phase 4: In this phase, the user computes $\mathbf{z}_i = \mathbf{y}_i + \mathbf{a}_i$ for all $i \in [l]$. The rejection sampling is used here to ensure that $\mathbf{z}_i$ is independent of $\mathbf{y}_i$ for blindness. If $\|\mathbf{z}_i\| \leq \eta\sqrt{m}\sigma_3$ for all $i \in [l]$ then the user outputs $(\mu, \Sigma = ((\mathbf{z}_i)_{i\in[l]}, \mathbf{c}, \mathbf{t}))$ as the *final signature*; otherwise, he returns "$\perp$". Note that, it is a must for the user to send result to the signer as a confirmation of the validity of the final signature (if $\mathsf{result} := \mathsf{accept}$) or as a requirement to restart the protocol (if $\mathsf{result} := ((\mathbf{a}_i)_{i\in[l]}, \mathbf{b}, \mathbf{c}, C)$).
- Phase 5: In this phase, if the signer gets $\mathsf{result} \neq \mathsf{accept}$, he will check up some conditions before he restarts the protocol from the beginning. This helps to detect the case that an adversarial user tries to restart the signing protocol despite having obtained a valid signature. If the signer gets the validity confirmation from the user, he finally outputs the *view* $\mathcal{V} = (\mathbf{x}, \mathbf{e}, (\mathbf{s}_i, \mathbf{y}_i)_{i\in[l]})$.

– $\mathsf{BRS.Verify}(\mathcal{P}, \mu, \Sigma, PK) = 1$ iff $\|\mathbf{z}_i\| \leq \eta\sqrt{m}\sigma_3$ for all $i \in [l]$ and $\mathbf{c} = H(\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{Tc} \pmod q, \mathsf{com}(\mu, \mathbf{t}), PK)$; and $\mathsf{BRS.Verify}(\mathcal{P}, \mu, \Sigma, PK) = 0$ otherwise. (See Algorithm 3.)

5 Correctness and Security Analysis of BRS

5.1 Correctness

Theorem 1 (Correctness). *Our BRS scheme is correct after at most e^2 repetitions with probability at least $1 - 2^{-100}$.*

Proof (of Theorem 1). Given the pair $(\mu, \Sigma = ((\mathbf{z}_i)_{i\in[l]}, \mathbf{c}, \mathbf{t}))$ is the output of the user in $\mathsf{BRS.Sign}(\mathcal{P}, \mathbf{S}_j, \mu, PK)$ as in Figure 1, the set of public keys $PK =$

Algorithm 3 BRS.Verify$(\mathcal{P}, \mu, \Sigma, PK)$

Input: $\mathcal{P}$, $\mu, \Sigma = ((\mathbf{z}_i)_{i\in[l]}, \mathbf{c}, \mathbf{t})$, $PK = \{\mathbf{A}_i\}_{i\in[l]}$
Output: 1 or 0
1: $\mathbf{u} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{Tc} \pmod q$
2: $\mathbf{c}' = H(\mathbf{u}, \mathsf{com}(\mu, \mathbf{t}), PK)$
3: **if** $\mathbf{c}' = \mathbf{c}$ and $\|\mathbf{z}_i\| \le \eta\sqrt{m}\sigma_3$ for all $i \in [l]$ **then**
4: **return** 1
5: **else**
6: **return** 0
7: **end if**

$\{\mathbf{A}_i\}_{i\in[l]}$, and parameters $\mathcal{P}$, we will prove that $H(\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{Tc} \pmod q, \mathsf{com}(\mu, \mathbf{t}), PK) = \mathbf{c}$.

Without caring the restarts appear in rejection samplings, we have

$$\begin{aligned}\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{Tc} \pmod q &= \sum_{i\in[l]} \mathbf{A}_i(\mathbf{y}_i + \mathbf{a}_i) - \mathbf{T}(\mathbf{e} - \mathbf{b}) \pmod q \\ &= \sum_{i\in[l]\setminus\{j\}} \mathbf{A}_i\mathbf{s}_i + \mathbf{A}_j(\mathbf{s}_j + \mathbf{S}_j\mathbf{e}) \\ &\quad + \sum_{i\in[l]} \mathbf{A}_i\mathbf{a}_i - \mathbf{T}(\mathbf{e} - \mathbf{b}) \pmod q \\ &= \sum_{i\in[l]} \mathbf{A}_i\mathbf{s}_i + \sum_{i\in[l]} \mathbf{A}_i\mathbf{a}_i + \mathbf{Tb} \pmod q \\ &= \mathbf{x} + \mathbf{w} + \mathbf{Tb} \pmod q.\end{aligned}$$

Hence $H(\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{Tc} \pmod q, \mathsf{com}(\mu, \mathbf{t}), PK) = \mathbf{c}$. Note that, with overwhelming probability, $\|\mathbf{z}_i| \le \eta\sqrt{m}\sigma_3$ for all $i \in [l]$ by Lemma 2.

Now we analyze the rejection sampling technique to bound the number of restarts of our BRS protocol. Recall that, by Remark 3, we have

$$\frac{\mathcal{D}_\sigma^m(\mathbf{x})}{M \cdot \mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{x})} \le \frac{e^{1+1/288}}{M},$$

with probability at least $1 - 2^{-100}$ if $\sigma = 12\|\mathbf{v}\|$. Being used in the rejection sampling, we need $\mathcal{D}_\sigma^m(\mathbf{x})/(M \cdot \mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{x})) \le 1$. Since M should be as small as possible, it is sufficient to choose $M = \exp\left((24\|\mathbf{v}\|\sigma + \|\mathbf{v}\|^2)/(2\sigma^2)\right) \approx e^{1+1/288}$, with $\sigma = 12\|\mathbf{v}\|$. Now we apply above analyses to the rejection samplings in our BRS scheme. Remark that in Phase 2 of our scheme, as the user utilizes the rejection sampling locally to output $\mathbf{e}$, the restarts of this phase does not impact to the correctness of the scheme. We just care about the restarts happening in Phase 3 and Phase 5. Hence, after at most $M_2 \cdot M_3 \approx e^2$ restarts, the BRS scheme can successfully output a valid blind ring signature. □

Remark 5. In the proof of Theorem 1, we use $\mathbf{e} = \mathbf{c} + \mathbf{b}$ obtained in Phase 2 of $\mathsf{BRS.Sign}$. Assume that $\mathbf{e} = \mathbf{c}' + \mathbf{b}'$ for some $\mathbf{b} \neq \mathbf{b}'$, $\mathbf{c} \neq \mathbf{c}'$, then also

$$\begin{aligned}\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{T}\mathbf{c}' \pmod q &= \sum_{i\in[l]} \mathbf{A}_i(\mathbf{y}_i + \mathbf{a}_i) - \mathbf{T}(\mathbf{e} - \mathbf{b}') \pmod q \\ &= \mathbf{x} + \mathbf{w} + \mathbf{T}\mathbf{b}' \pmod q.\end{aligned}$$

Thus, if $\mathbf{e}-\mathbf{b}' = \mathbf{c}' = H(\mathbf{x}+\mathbf{w}+\mathbf{T}\mathbf{b}' \pmod q, \mathsf{com}(\mu, \mathbf{t}), PK)$, then $H(\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i - \mathbf{T}\mathbf{c}' \pmod q, \mathsf{com}(\mu, \mathbf{t}), PK) = \mathbf{c}'$. This remark will be used in the proof of Theorem 4.

5.2 Anonymity

Recall that, in the anonymity game (see Subsection 3), the adversary $\mathcal{A}$ receives a set of public keys $PK = \{\mathsf{pk}_i\}_{i\in[l]}$ and he adaptively make queries to the blind ring signature with a message μ and the indexes j, $i_1 \in [l]$ to get a signature Σ which depends on the random bit $b \in \{0, 1\}$ chosen by the challenger. The adversary wins the game if he guesses exactly the bit b. The following theorem says that the advantage of the attacker in guessing b is actually negligible.

Theorem 2 (Anonymity). *Given the ring of signers $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$, and the set of key pairs $\{(\mathbf{A}_i, \mathbf{S}_i)\}_{i\in[l]}$, a message μ, two distinct indexes j, $i_1 \in [l]$ and a random bit $b \in \{0, 1\}$. Consider the anonymity game as in Subsection 3. Let X_0 and X_1 two random variables representing the blinded signatures obtained by the blind ring signature protocol $\mathsf{BRS.Sign}$ with respect to $b = 0$ and $b = 1$, respectively. Then there exist a universal constant $M_2 > 0$ such that*

$$\Delta(X_0, X_1) \leq \frac{2^{1-\omega(\log m)}}{M_2}.$$

Proof (of Theorem 2). In the game, the challenger chooses randomly $b \in \{0, 1\}$ and runs $\mathsf{BRS.Sign}$ using the signer $\mathcal{S}_{i_b}$ corresponding to the private key $\mathbf{S}_{i_b}$, then we will get the blinded signature $(\mathbf{y}_1, \cdots, \mathbf{y}_{i_b} \cdots, \mathbf{y}_l)$, where $\mathbf{y}_{i_b} := \mathbf{S}_{i_b}\mathbf{e} + \mathbf{s}_{i_b}$ outputted with probability $\min\{\mathcal{D}^m_{\sigma_2}(\mathbf{y}_{i_b})/(M_2 \cdot \mathcal{D}^m_{\mathbf{S}_{i_b}\mathbf{e},\sigma_2}(\mathbf{y}_{i_b})), 1\}$ and $\mathbf{y}_i := \mathbf{s}_i \leftarrow \mathcal{D}^m_{\sigma_2}$ for all $i \in [l] \setminus \{i_b\}$.

Assume that the adversary gets the signature $(\mathbf{y}_1, \cdots, \mathbf{y}_{i_b} \cdots, \mathbf{y}_l)$ by choosing each element $\mathbf{y}_i$ from $\mathcal{D}^m_{\sigma_2}$ with probability $1/M_2$. We denote by Y the random variable according to the signature obtained by this way. Then using Lemma 5 we have

$$\Delta(X_0, Y) \leq \frac{2^{-\omega(\log m)}}{M_2} \text{ and } \Delta(X_1, Y) \leq \frac{2^{-\omega(\log m)}}{M_2}.$$

Hence $\Delta(X_0, X_1) \leq \Delta(X_0, Y) + \Delta(X_1, Y) \leq \frac{2^{1-\omega(\log m)}}{M_2}$ still negligible. □

5.3 Blindness

Theorem 3 (Blindness). *Our BRS scheme is blind provided that* com *is hiding and the hash function H is one-way and collision-resistant.*

Proof (of Theorem 3). It is easy to see that the blindness of our BRS scheme is guaranteed by the rejection sampling technique and the hiding property of the commitment com.

As per the game of blindness in Subsection 3, when the dishonest signer gives two messages μ_0 and μ_1 to the challenger, the challenger will chooses randomly a bit $b \in \{0, 1\}$. Then the signer and the challenger initiates the blind ring signature protocol having interaction with only one of two users $\mathcal{U}(\mathcal{P}, PK, \mu_0)$ and $\mathcal{U}(\mathcal{P}, PK, \mu_1)$. We show that the signer actually does not know which user he is interacting with, that is, the view $\mathcal{V} = (\mathbf{x}, \mathbf{e}, (\mathbf{s}_i, \mathbf{y}_i)_{i\in[l]})$ that the signer has is independent of the message being signed. More precisely, $\mathbf{e}$ and $(\mathbf{y}_i)_{i\in[l]})$ is independent of the message being signed. Indeed, let $\mathcal{V}_0 = (\mathbf{x}_0, \mathbf{e}_0, (\mathbf{s}_{0,i}, \mathbf{y}_{0,i})_{i\in[l]})$ and $\mathcal{V}_1 = (\mathbf{x}_1, \mathbf{e}_1, (\mathbf{s}_{1,i}, \mathbf{y}_{1,i})_{i\in[l]})$ be views respectively corresponding to users $\mathcal{U}(\mathcal{P}, PK, \mu_0)$ and $\mathcal{U}(\mathcal{P}, PK, \mu_1)$. Then, the rejection sampling in Phase 2 ensures that both $\mathbf{e}_0$ and $\mathbf{e}_1$ are distributed according to the same distribution $\mathcal{D}^k_{\sigma_1}$. Similarly, by the rejection sampling in Phase 3, both $\mathbf{y}_{0,i}$ and $\mathbf{y}_{1,j}$ for all $i, j \in [l]$ follow the same distribution $\mathcal{D}^m_{\sigma_2}$. The distributions of $\mathbf{e}$ and $\mathbf{y}_{0,i}$ are independent of choosing the message to be signed.

Regarding two unblinded signatures $\Sigma_0 = ((\mathbf{z}_{b,i})_{i\in[l]}, \mathbf{c}_b, \mathbf{t}_b)$ corresponding to the users $\mathcal{U}(\mathcal{P}, PK, \mu_b)$, $b = 0, 1$. Again, by the rejection sampling used in Phase 4, the malicious signer is impossible to distinguish $(\mathbf{z}_{0,i})_{i\in[l]}$ from $(\mathbf{z}_{1,i})_{i\in[l]}$. Certainly, the signer does not learn anything about the original message μ being signed from the challenges $\mathbf{c}_0$, $\mathbf{c}_1$ due to the property of the hash function H. Also, the distribution of $\mathbf{t}_b$ is independent of μ.

Finally, we concern the restart might happen in Phase 5. Again, by the hiding property of the commitment com and since the user samples fresh values $\mathbf{t}$, $\mathbf{a}$ and $\mathbf{b}$ after every such a restart, we have that each rerun of the protocol is independent of the previous runs. (See similar arguments to a blind signature scheme in [20].) □

5.4 One-more Unforgeability

Before stating the main theorem of this subsection, we adopt the following lemma:

Lemma 6 (Lemma 5.2 in [15]). *Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ where $m > 64 + n \log q / \log(2d+1)$, randomly chosen $\mathbf{s} \leftarrow_{\$} \{-d, \cdots, 0, \cdots, d\}^m$. Then with probability at least $1 - 2^{-100}$, there exists another $\mathbf{s}' \leftarrow_{\$} \{-d, \cdots, 0, \cdots, d\}^m$ such that $\mathbf{As} = \mathbf{As}' (mod\ q)$.*

For notational convenience, we call the *(q_H, q_S, δ)-forger* $\mathcal{F}$ a polynomial-time algorithm $\mathcal{F}$ that successfully breaks the one-more unforgeablity of our BRS protocol with non negligible probability δ, making at most q_H hash queries and

at most q_S sign queries to the scheme. The following theorem says that if there exists such a forger then one can construct an algorithm being able to solve an SIS problem.

Theorem 4 (One-more Unforgeability). *Consider the* BRS *scheme described in Section 4. Suppose that the commitment function* com *used in the* BRS *scheme is binding. If there is a* (q_H, q_S, δ)*-forger* $\mathcal{F}$ *who breaks the one-more unforgeablity of our* BRS *protocol then there is a polynomial-time algorithm* $\mathcal{G}$ *which can solve an* $\mathsf{SIS}_{q,n,ml,\beta}$ *problem with* $\beta = \max\{(2l\eta\sigma_3 + 2\sigma\sqrt{\kappa})\sqrt{m}, (2l\eta\sigma_3 + l\eta\sigma_2)\sqrt{m}\}$ *with probability at least*

$$\delta_{overall} \geq \min\left\{\frac{1}{4s}(1-\zeta)\left(1-\frac{1}{|D_c|}\right)\left(\frac{\delta-\frac{1}{|D_c|}}{q_H}-\frac{1}{|D_c|}\right), \delta\left(1-\frac{1}{|D_c|}\right)\right\},$$

where ζ *is the probability of a restart in the scheme,* $s := q_S + 1$.

Proof (of Theorem 4).

In the following, we will describe an algorithm $\mathcal{G}$ using $\mathcal{F}$ as a black-box routine to solve the following $\mathsf{SIS}_{q,n,ml,\beta}$ problem:

$$\text{Find } \|\widehat{\mathbf{z}}\| \leq \beta \text{ such that } \mathbf{A}\widehat{\mathbf{z}} = \mathbf{0} \pmod{q}, \tag{2}$$

where $\mathbf{A} := [\mathbf{A}_1\|\cdots\|\mathbf{A}_l]$, and all $\mathbf{A}_i$'s are random matrices in $\mathbb{Z}_q^{n\times m}$.

1. **Setup.** First of all, $\mathcal{G}$ calls $\mathsf{BRS.Setup}(1^n)$ to get a set of public parameters $\mathcal{P}$, but without $\mathbf{T}$. This $\mathbf{T}$ will be produced by $\mathcal{G}$ later, as below. $\mathcal{G}$ then forms a set of signers to generate the ring of signers $R = \{\mathcal{S}_1, \cdots, \mathcal{S}_l\}$ in which each signer $\mathcal{S}_i$ is uniquely identified by the matrix $\mathbf{A}_i$. Next, $\mathcal{G}$ chooses randomly an index $j \leftarrow_\$ \{1, \cdots, l\}$ in order for $\mathcal{G}$, when necessary (e.g., to reply signing queries made by $\mathcal{F}$), to play the role of $\mathcal{S}_j$. Afterwards, $\mathcal{G}$ samples $\mathbf{S}_j \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m\times k}$ and sets $\mathbf{T} := \mathbf{A}_j\mathbf{S}_j \pmod{q}$ and adds $\mathbf{T}$ to $\mathcal{P}$. Finally, $\mathcal{G}$ sends to the forger $\mathcal{F}$ the set $\mathcal{P}$ and the set of public keys $PK := \{\mathbf{A}_i\}_{i\in[l]}$. The matrix $\mathbf{S}_j$ will be kept as a secret key.
2. **Queries.** The forger $\mathcal{F}$ adaptively makes q_H hash queries to the random oracle which models the hash function H in the real protocol and q_S blind signing queries. The algorithm $\mathcal{G}$ creates and maintains a list L_H consisting of random oracle queries $(\mathbf{u}, C) \leftarrow_\$ \mathbb{Z}_q^n \times \{0,1\}^n$ and their corresponding hash value $\mathbf{c} \in D_c$, where $D_c := \{\mathbf{c} : \mathbf{c} \in \{-1, 0, 1\}^k, \|\mathbf{c}\|_1 \leq \kappa\}$. Furthermore, $\mathcal{G}$ randomly preselects $\mathcal{R} := \{\mathbf{r}_1, \cdots, \mathbf{r}_{q_H}\} \leftarrow_\$ D_c$ as a set of replies of H and also chooses a random tape ρ. The solver $\mathcal{G}$ runs $\mathcal{F}(\mathcal{P}, PK, \rho)$ as a black-box routine as follows:
   - **Random Oracle Queries.** Whenever $\mathcal{G}$ receives a query $(\mathbf{u}, C)$, it will check whether the query is in the list L_H or not. If yes, $\mathcal{G}$ sends the corresponding hash value $\mathbf{c}$ to the forger $\mathcal{F}$. Otherwise, $\mathcal{G}$ opts the first unused $\mathbf{r}_i, i \in [q_H]$ from $\mathcal{R}$, assigns $\mathbf{c} := \mathbf{r}_i$, stores the query-hash value pair $((\mathbf{u}, C), \mathbf{c})$ in L_H and sends $\mathbf{c}$ to the forger.

– **Signing Queries.** The forger $\mathcal{F}$ plays the role of the user, processing q_S times the interactive blind ring signature protocol, while the solver $\mathcal{G}$ acts as the signer of the ring. If $\mathcal{F}$ wants to have the signature of a message μ, the solver $\mathcal{G}$ will plays the role of the signer $\mathcal{S}_j$ and runs the BRS.Sign algorithm in Figure 1 to produce the required signature using the matrix $\mathbf{S}_j$ as the secret key in Phase 3.

3. **Output.** After at most q_S signing queries, with non-negligible probability δ, the forger $\mathcal{F}$ eventually outputs $s := q_S + 1$ blind ring signatures

$$(\mu_1, (\mathbf{z}_{1,i})_{i\in[l]}, \mathbf{c}_1, \mathbf{t}_1), \cdots, (\mu_s, (\mathbf{z}_{s,i})_{i\in[l]}, \mathbf{c}_s, \mathbf{t}_s),$$

where $\mu_1, \cdots, \mu_s$ are s distinct messages. At the moment, the algorithm $\mathcal{G}$ predicts randomly an index $k \in [s]$ satisfying that $\mathbf{c}_k = \mathbf{r}_i$ for some $i \in [q_H]$. Afterward, $\mathcal{G}$ samples new fresh random oracle answers $\{\mathbf{r}'_i, \cdots, \mathbf{r}'_{q_H}\} \leftarrow_{\$} D_c$ and then invokes $\mathcal{F}(\mathcal{P}, PK, \rho)$ again with $\mathcal{R}' := \{\mathbf{r}_1, \cdots, \mathbf{r}_{i-1}, \mathbf{r}'_i, \cdots, \mathbf{r}'_{q_H}\}$. Among other values, the forger $\mathcal{F}$ outputs $(\mu'_k, (\mathbf{z}'_{k,i})_{i\in[l]}, \mathbf{c}'_k, \mathbf{t}'_k)$. If $\mathbf{c}_k \neq \mathbf{c}'_k$ then $\mathcal{G}$ returns

$$((\mathbf{z}_{k,i})_{i\in[l]} - \mathbf{S}_j\mathbf{c}_k, (\mathbf{z}'_{k,i})_{i\in[l]} - \mathbf{S}_j\mathbf{c}'_k) \text{ for all } j \in [l],$$

in order to solve the SIS problem. If $\mathbf{c}_k = \mathbf{c}'_k$, the solver $\mathcal{G}$ retries $\mathcal{F}(\mathcal{P}, PK, \rho')$ at most q_H^s times with a different random tape ρ'.

Analysis. The environment of $\mathcal{F}$ is perfectly simulated by $\mathcal{G}$ since the distribution of the matrix $\mathbf{T}$, which is generated by sampling $\mathbf{S}_j \leftarrow \mathcal{D}^{m\times k}_{\mathbb{Z},\sigma}$ with sufficiently large σ then and computing $\mathbf{T} := \mathbf{A}_j\mathbf{S}_j \pmod q$, is close to uniform (see [8, lemma 5.2] for more details). Moreover, rejection sampling is exploited before outputting $\mathbf{y}_j$ (in Phase 3) then $\mathbf{y}_j$ is independent of $\mathbf{S}_j$, thus $\mathcal{F}$ learns no information about $\mathbf{S}_j$ from receiving $\mathbf{y}_j$. Therefore, the restarts happen with the same probability ζ as in the real scheme. Obviously, there is at least one signature not coming from a real interaction. The algorithm $\mathcal{G}$ guesses correctly the index of this signature with probability at least $1/s$. And $\mathbf{c}_k$ is a random oracle answer with probability $1/|D_c|$. Note that, with probability $1/2$, there is at least one of the reruns of $\mathcal{F}$ gives the same index pair (i, k) such that $\mathbf{r}_i = \mathbf{c}_k$. Therefore, we can assume that the index pairs in two runs are the same.

Applying the forking lemma [3, Lemma 3.1] with noting that restarts happen with probability ζ, we have that $\mathcal{F}$ is again successful in breaking the one-more unforgeability and outputs one more new signature $(\mu'_k, (\mathbf{z}'_{k,i})_{i\in[l]}, \mathbf{c}'_k, \mathbf{t}'_k)$ with probability $\delta_{frk} \geq (1-\zeta)(\delta - 1/|D_c|)((\delta - 1/|D_c|)/q_H - 1/|D_c|)$ using the same random oracle query as in the first run. Thus we have

$$(\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_{k,i} - \mathbf{T}\mathbf{c}_k \pmod q, \mathsf{com}(\mu_k, \mathbf{t}_k)) = (\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}'_{k,i} - \mathbf{T}\mathbf{c}'_k \pmod q, \mathsf{com}(\mu'_k, \mathbf{t}'_k)).$$

Since then, we have that

$$\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_{k,i} - \mathbf{T}\mathbf{c}_k \pmod q = \sum_{i\in[l]} \mathbf{A}_i\mathbf{z}'_{k,i} - \mathbf{T}\mathbf{c}'_k \pmod q.$$

Equivalently,

$$\sum_{i\in[l]} \mathbf{A}_i(\mathbf{z}_{k,i} - \mathbf{z}'_{k,i}) + \mathbf{T}(\mathbf{c}'_k - \mathbf{c}_k) = \mathbf{0} \pmod q. \tag{3}$$

Plugging $\mathbf{T} = \mathbf{A}_j\mathbf{S}_j \pmod q$ into Eq. (3), we have

$$\sum_{i\in[l]\setminus\{j\}} \mathbf{A}_i(\mathbf{z}_{k,i} - \mathbf{z}'_{k,i}) + \mathbf{A}_j(\mathbf{z}_{k,j} - \mathbf{z}'_{k,j} + \mathbf{S}_j(\mathbf{c}'_k - \mathbf{c}_k)) = \mathbf{0} \pmod q. \tag{4}$$

Set the matrix

$$\mathbf{A} := [\mathbf{A}_1\|\cdots\|\mathbf{A}_{j-1}\|\mathbf{A}_j\|\mathbf{A}_{j+1}\|\cdots\|\mathbf{A}_l],$$

and

$$\begin{aligned}\widehat{\mathbf{z}} :=&[\mathbf{z}_{k,1} - \mathbf{z}'_{k,1}, \cdots, \mathbf{z}_{k,j-1} - \mathbf{z}'_{k,j-1}, \mathbf{z}_{k,j} - \mathbf{z}'_{k,j} + \mathbf{S}_j(\mathbf{c}'_k - \mathbf{c}_k),\\ &\mathbf{z}_{k,j+1} - \mathbf{z}'_{k,j+1}, \cdots, \mathbf{z}_{k,l} - \mathbf{z}'_{k,l}],\end{aligned}$$

from Equation (4) we have $\mathbf{A}\widehat{\mathbf{z}} = \mathbf{0} \pmod q$.

The next step is to prove that $\widehat{\mathbf{z}} \neq \mathbf{0}$ with probability non- negligible. In fact, by Lemma 6, there is another secret key $\mathbf{S'}_j$ such that $\mathbf{AS}_j = \mathbf{AS}'_j \pmod q$ in which $\mathbf{S}_j$ and $\mathbf{S}_j$ have all the same columns but the i-th column with i is the position that $\mathbf{c}_k[i] \neq \mathbf{c}'_k[i]$. Clearly, if $\mathbf{z}_{k,j} - \mathbf{z}'_{k,j} + \mathbf{S}_j(\mathbf{c}'_k - \mathbf{c}_k) = \mathbf{0}$ then $\mathbf{z}_{k,j} - \mathbf{z}'_{k,j} + \mathbf{S}'_j(\mathbf{c}'_k - \mathbf{c}_k) \neq \mathbf{0}$. Thus with probability at least $1/2$ we get $\widehat{\mathbf{z}} \neq 0$. Note that $\|\mathbf{z}_{k,i}\| \leq \eta\sigma_3\sqrt{m}$, $\|\mathbf{S}_i\| \leq \sigma\sqrt{m}$ and $\|\mathbf{c}_k\| \leq \sqrt{\kappa}$ for all $i \in [l]$. Hence, $\|\mathbf{z}_{k,i} - \mathbf{z'}_{k,i}\| \leq 2\eta\sigma_3\sqrt{m}$ for all $i \in [l]$,. Thus, $\|\widehat{\mathbf{z}}\| \leq (2l\eta\sigma_3 + 2\sigma\sqrt{\kappa})\sqrt{m}$. Therefore, we have the success probability of $\mathcal{G}$ in solving the SIS problem (2) in this case is at least

$$\delta_{solve} \geq \frac{1}{4s}\delta_{frk} \geq \frac{1}{4s}(1-\zeta)(\delta - 1/|D_c|)((\delta - 1/|D_c|)/q_H - 1/|D_c|).$$

Now, taking restarts happen in Phase 5 into account, we will show that if the adversarial user can forge a valid signature through a Phase 5 restart help, then $\mathcal{G}$ can solve the SIS problem stated in Equation (2). To trigger a restart in Phase 5, the forger sends to the signer $\mathsf{result}:= ((\mathbf{a}_i)_{i\in[l]}, \mathbf{b}, \mathbf{c}, C)$ which, together with the view of the signer $\mathcal{V} = (\mathbf{x}, \mathbf{e}, (\mathbf{s}_i, \mathbf{y}_i)_{i\in[l]})$, satisfies all the following conditions:

$$\mathbf{e} - \mathbf{b} = \mathbf{c} = H(\mathbf{x} + \mathbf{w} + \mathbf{Tb} \pmod q, C, PK), \tag{5}$$

$$\mathbf{c} = H(\mathbf{w} + \mathbf{v} - \mathbf{Tc} \pmod q, C, PK), \tag{6}$$

$$\|\mathbf{y}_j + \mathbf{a}_j\| > \eta\sigma_3\sqrt{m} \text{ for some } j \in [l], \tag{7}$$

where $\mathbf{x} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{s}_i$, $\mathbf{w} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{a}_i$, $\mathbf{v} = \sum_{i\in[l]} \mathbf{A}_i\mathbf{y}_i$. Assume that the adversary can obtain a valid signature $\Sigma^* = ((\mathbf{z}'_i)_{i\in[l]}, \mathbf{c}', \mathbf{t}')$ (with probability at least δ) from this restart. That is, for some $\mathbf{b}' \in \mathcal{D}^k_{\sigma_1}$ such that $\mathbf{e} = \mathbf{c}' + \mathbf{b}'$ we

have,

$$\mathbf{e} - \mathbf{b}' = \mathbf{c}' = H(\mathbf{x} + \mathbf{w} + \mathbf{T}\mathbf{b}' \pmod q, C, PK), \tag{8}$$

$$\mathbf{c}' = H(\sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i' - \mathbf{T}\mathbf{c}' \pmod q, \mathsf{com}(\mu, \mathbf{t}'), PK), \tag{9}$$

$$\|\mathbf{z}_i'\| \leq \eta\sigma_3\sqrt{m} \text{ for all } i \in [l]. \tag{10}$$

With probability $1-1/|D_c|$ (how to compute this probability, see [10, Subsection 4.6.1 in Chapter 4])), we have $\mathbf{c}' = \mathbf{c}$. Then by Equation (6) and Equation (9), we have

$$\mathbf{w} + \mathbf{v} \pmod q = \sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i' \pmod q.$$

That is,

$$\sum_{i\in[l]} \mathbf{A}_i(\mathbf{a}_i + \mathbf{y}_i) \pmod q = \sum_{i\in[l]} \mathbf{A}_i\mathbf{z}_i' \pmod q.$$

Define

$$\widehat{\mathbf{z}} := [\mathbf{a}_1 + \mathbf{y}_1 - \mathbf{z}_1', \cdots, \mathbf{a}_l + \mathbf{y}_l - \mathbf{z}_l'].$$

We have $\mathbf{A}\widehat{\mathbf{z}} = \mathbf{0} \pmod q$. If $\widehat{\mathbf{z}} = 0$, i.e., $\mathbf{a}_i + \mathbf{y}_i = \mathbf{z}_i'$ for all $i \in [l]$, then we have $\|\mathbf{y}_i + \mathbf{a}_i\| \leq \eta\sigma_3\sqrt{m}$ for all $i \in [l]$ (due to Equation (10)) which contradicts with Equation (7). Hence, $\widehat{\mathbf{z}} \neq \mathbf{0}$ and we have $\|\widehat{\mathbf{z}}\| \leq (2l\eta\sigma_3 + l\eta\sigma_2)\sqrt{m}$. The success probability of $\mathcal{G}$ in case the forger can get a valid signature through a restart is $\delta_{restart} \geq \delta(1 - 1/|D_c|)$.

To sum up, we have proven that with overall success probability of $\delta_{overall} \geq \min(\delta_{solve}, \delta_{restart})$, the solver $\mathcal{G}$ can solve the $\mathsf{SIS}_{q,n,ml,\beta}$ problem where

$$\beta = \max((2l\eta\sigma_3 + 2\sigma\sqrt{\kappa})\sqrt{m}, (2l\eta\sigma_3 + l\eta\sigma_2)\sqrt{m}).$$

6 Parameter Setting

Basically, parameters in this work are set in a similar way to [27]. We need parameters n, q, k to make sure that the SIS problem is computationally infeasible to keep secret keys $\mathbf{S}_i$'s not to be recovered. To generate the key pairs, we invoke the trapdoor functions using the discrete Gaussian distribution D_σ with $\sigma \geq L \cdot \omega(\sqrt{\log n})$ and $L = m^{1+\epsilon}$ for any $\epsilon > 0$.

For security proofs, we need $m \geq 64 + n\log q/\log(2d+1)$ via Lemma 6. We also need $m \geq 5n\log q$ for TrapGen works. So we can choose $m \geq \max\{64 + n \cdot \log q/\log(2d+1), 5n\log q\}$. The parameter κ appearing in the hash function H should be chosen to satisfy $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$ in order to guarantee that the min-entropy of H is at least 100. As analyzed in Subsection 5.1, we can set $M_i := e^{1+1/288}$ for all $i \in [3]$. Accordingly, we then set $\sigma_1 = 12\|\mathbf{c}\| = 12\sqrt{\kappa}$, $\sigma_2 = 12\|\mathbf{S}_j\mathbf{e}\| = 12\sigma\eta\sigma_1\sqrt{mk} = 144\sigma\eta\sqrt{mk\kappa}$ and $\sigma_3 = 12\|\mathbf{y}_i\| = 12\eta\sigma_2\sqrt{m} = 1728m\eta^2\sigma\sqrt{k\kappa}$.

Parameters	Requirement	Description
n	–	security parameter
l	–	number of ring members
q	$poly(n)$, prime	modulo
m	$\max(64 + n\log q/\log(2d+1), 5n\log q)$	in Lemma 6, TrapGen
K	$m^{1+\epsilon}$, for any $\epsilon > 0$	in SampleKey
σ	$\geq K \cdot \omega(\sqrt{\log n})$	in SampleKey
d	$\sigma \cdot \sqrt{m}$	in BRS.KeyGen
k and κ	such that $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$	in the hash function H
η	[1.1, 1.3]	in Lemma 2
$M_1 = M_2 = M_3$	$\exp(1 + 1/288)$	in the rejection sampling
σ_1	$12\sqrt{\kappa}$	
σ_2	$12\sigma\eta\sigma_1\sqrt{mk}$	
σ_3	$12\eta\sigma_2\sqrt{m}$	
signature size	$lm\log(12\sigma_3) + n + \kappa$ bits	
secret key size	$lmk\log(2d+1)$ bits	
public key size	$(lnm + nk)\log q$ bits	

Table 2: Parameter setting for our BRS scheme

The real signature is $\Sigma = ((\mathbf{z}_i)_{i\in[l]}, \mathbf{c}, \mathbf{t}))$. Each component of $\mathbf{z}_i$ is of length at most $12\sigma_3$ with probability at least $1 - 2^{-100}$ by Lemma 1, so the signature bit-size is $lm\log(12\sigma_3) + n + \kappa$ bits. The secret key bit-size is $lmk\log(2d+1)$. The public key bit-size is $(lnm + nk)\log q$.

The parameter setting is summarized in Table2.

7 Conclusions and Future Works

In this paper, we proposed, for the first time, a lattice-based blind ring signature scheme. Our scheme is proven to fulfill the anonymity and the blindness properties due to being constructed with the reject sampling technique . Moreover, the scheme is one-more unforgeable in the random oracle model under the hardness of SIS problem.

There have been several recent results in improving lattice-based (ring) signatures both on signature sizes and the hardness assumption (e.g. from module lattices), which can be utilized to improve our scheme. We will leave to apply these improvements as future works. One more interesting approach should be our next work is to design a blind ring signature without using trapdoor functions but, for example, basing on the idea of [1]. Also, it is still open to construct a blind ring signature that is secure in the standard model.

Acknowledgment. The first author would like to thank Prof. Masaya Yasuda for his financial support. The authors would like to thank anonymous reviewers for their helpful comments.

References

1. Aguilar-Melchor, C., Bettaieb, S., Boyen, X., Fousse, L., Gaborit, P.: Adapting Lyubashevsky's Signature Schemes to the Ring Signature Setting. Cryptology ePrint Archive, Report 2013/281 (2013), https://eprint.iacr.org/2013/281
2. Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Abstract). In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. pp. 99–108. STOC '96, ACM, New York, NY, USA (1996), http://doi.acm.org/10.1145/237814.237838
3. Bellare, M., Neven, G.: New Multi-Signature Schemes and a General Forking Lemma. Full version, available from, http://soc1024.ece.illinois.edu/teaching/ece498ac/fall2018/forkinglemma.pdf
4. Bernstein, D.J., Buchmann, J., Dahmen, E.: Post Quantum Cryptography. Springer Publishing Company, Incorporated, 1st edn. (2008)
5. Bresson, E., Stern, J., Szydlo, M.: Threshold Ring Signatures and Applications to Ad-hoc Groups. In: Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. pp. 465–480 (2002), https://doi.org/10.1007/3-540-45708-9_30
6. Chaum, D.: Blind Signatures for Untraceable Payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology. pp. 199–203. Springer US, Boston, MA (1983)
7. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. pp. 186–194 (1986), https://doi.org/10.1007/3-540-47721-7_12
8. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for Hard Lattices and New Cryptographic Constructions. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing. pp. 197–206. STOC '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1374376.1374407
9. Ghadafi, E.M.: Sub-linear Blind Ring Signatures without Random Oracles. In: Stam, M. (ed.) Cryptography and Coding. pp. 304–323. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
10. Guo, F., Susilo, W., Mu, Y.: Foundations of Security Reduction, pp. 29–146. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-93049-7_4, https://doi.org/10.1007/978-3-319-93049-7_4
11. Halevi, S., Micali, S.: Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. In: Koblitz, N. (ed.) Advances in Cryptology — CRYPTO '96. pp. 201–215. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
12. Herranz, J., Laguillaumie, F.: Blind Ring Signatures Secure Under the Chosen-Target-CDH Assumption. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) Information Security. pp. 117–130. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
13. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In: Pieprzyk, J. (ed.) Advances in Cryptology - ASIACRYPT 2008. pp. 372–389. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
14. Lyubashevsky, V.: Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In: Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings. pp. 598–616 (2009), https://doi.org/10.1007/978-3-642-10366-7_35

15. Lyubashevsky, V.: Lattice Signatures Without Trapdoors. Cryptology ePrint Archive, Report 2011/537, Full version of paper appearing at Eurocrypt 2012, last revised 18 Oct 2017 (2012), https://eprint.iacr.org/2011/537
16. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a Cryptographic Perspective, The Kluwer International Series in Engineering and Computer Science, vol. 671. Kluwer Academic Publishers, Boston, Massachusetts (Mar 2002)
17. Micciancio, D., Regev, O.: Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM J. Comput. **37**(1), 267–302 (Apr 2007), http://dx.doi.org/10.1137/S0097539705447360
18. Peikert, C.: A Decade of Lattice Cryptography. Found. Trends Theor. Comput. Sci. **10**(4), 283–424 (Mar 2016), http://dx.doi.org/10.1561/0400000074
19. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. pp. 552–565 (2001), https://doi.org/10.1007/3-540-45682-1_32
20. Rückert, M.: Lattice-based Blind Signatures. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010. pp. 413–430. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
21. van Saberhagen, N.: Cryptonote v 2.0 (2013), https://cryptonote.org/whitepaper.pdf
22. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (Nov 1994), https://doi.org/10.1109/SFCS.1994.365700
23. Susilo, W., Mu, Y.: Non-interactive Deniable Ring Authentication. In: Information Security and Cryptology - ICISC 2003, 6th International Conference, Seoul, Korea, November 27-28, 2003, Revised Papers. pp. 386–401 (2003), https://doi.org/10.1007/978-3-540-24691-6_29
24. Susilo, W., Mu, Y., Zhang, F.: Perfect Concurrent Signature Schemes. In: Information and Communications Security, 6th International Conference, ICICS 2004, Malaga, Spain, October 27-29, 2004, Proceedings. pp. 14–26 (2004), https://doi.org/10.1007/978-3-540-30191-2_2
25. Wang, S., Zhao, R., Zhang, Y.: Lattice-based ring signature scheme under the random oracle model. International Journal of High Performance Computing and Networking **11**(4), 332–341 (2018), https://doi.org/10.1504/IJHPCN.2018.093236
26. Wu, Q., Zhang, F., Susilo, W., Mu, Y.: An Efficient Static Blind Ring Signature Scheme. In: Won, D.H., Kim, S. (eds.) Information Security and Cryptology - ICISC 2005. pp. 410–423. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
27. Zhang, P., Jiang, H., Zheng, Z., Hu, P., Xu, Q.: A New Post-Quantum Blind Signature From Lattice Assumptions. IEEE Access **6**, 27251–27258 (2018), https://doi.org/10.1109/ACCESS.2018.2833103