IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 22, 2019, accepted December 31, 2019, date of publication January 7, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964686

A New Constant-Size Group Signature Scheme From Lattices

QIN LUO[1] AND CHUN-YANG JIANG[2]
[1]School of Mathematical Sciences, Fudan University, Shanghai 200433, China
[2]Mathematics and Institute of Mathematics, Jilin University, Changchun 130012, China

Corresponding author: Qin Luo (qluo14@fudan.edu.cn)

ABSTRACT A lattice-based group signature scheme (LGSS) is an active cryptographic primitive, where each group member can sign messages anonymously in the name of the entire group and each valid signature should be traced to some group member on the lattice. In each LGSS, the size of the group signature usually depends on the number of group members and the security parameter. Thus, designing a constant-size LGSS is an interesting problem. At PKC 2018, Ling, Nguyen, Wang and Xu presented the first constant-size group signature scheme under lattice assumptions. Its design is based on a zero-knowledge argument of the knowledge of a valid message-signature pair for the Ducas-Micciancio signature scheme, which follows the sign-then-encrypt-then-prove protocol. In contrast to this work, we construct a new constant-size LGSS. The scheme adopts the sign-hybrid-encrypt approach and makes use of the Lyubashevsky signature scheme. Our work is efficient in the signing algorithm, more precise on the open algorithm and shorter in public key, secret key and signature size than previous studies. Furthermore, we prove that the scheme has full anonymity and full traceability under the Ring Learning With Errors and Ring Short Integer Solution assumptions in the random oracle model.

INDEX TERMS Group signatures, lattices, Lyubashevsky signature scheme, ring learning with errors, ring short integer solution.

I. INTRODUCTION

The group signature introduced by Chaum and Van Heyst in [1], is an important cryptographic concept. In each group signature scheme, users can sign messages on behalf of the group anonymously, because the resulting signature does not reveal the signer's identity. Moreover, the resulting message/signature pair is traceable when necessary, in the sense that users are kept accountable for the message/signature pair that they issue. These two appealing features make the group signature useful in many real-life applications, for instance, in trusted computing platforms, auction protocols or privacy-protecting mechanisms, and digital rights management.

In the group signature realm, the signature size has been the focus of research for decades. Generally, the group signature size depends on the size N of the group, in addition to the security parameter. In early studies [1]–[3], the signature size grew linearly with N. The first approach in which the signature size was independent of N, and hence could be considered as a constant, was proposed in [4], and was later extended in [5]–[7].

To be precise, these group signature schemes [5]–[7] were built on the discrete logarithm assumption. With the development of post-quantum cryptography, the lattice-based group signature has become a research topic of great interest. In 2010, Gordon *et al.* [8] proposed the first group signature scheme based on lattice assumptions in the random oracle model, and its signature size is linear in N. Other lattice-based models were proposed later [20], [22], [26], and their signature sizes always depends on N. At PKC 2018, Ling *et al.* [9] constructed the first constant-size group signature scheme from lattices.

In the work of Ling *et al.*, the core of the design is based on a zero-knowledge argument of the knowledge of a valid message-signature pair for the Ducas-Micciancio signature scheme (DMS) [10]. A similar protocol for the Boyen signature scheme [38] was proposed by Ling *et al.* [22]. There exists a natural problem of whether other efficient signature schemes can replace the DMS and Boyen signature scheme. Based on this, the Lyubashevsky signature scheme (LSS) [13] has drawn our attention, and is the first lattice-based

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Farouk.

signature scheme without a trapdoor. In the LSS, Lyubashevsky removed the hash-and-sign model and proposed rejection sampling. Because of the small signature size and simple signing algorithm, the LSS is popular in latticed signature research. There has been much recent progress on the Lyubashevsky model in terms of its security, efficiency, and performance, such as [14]–[18]. These studies inspired us to investigate the problem of designing a constant-size latticed-based group signature scheme (LGSS) using the LSS.

A. RELATED WORKS

The first LGSS was introduced by Gordon, Katz and Vaikuntanathan, and its solution produced signature size linear in N. Camenisch *et al.* [19] extended [8] to achieve anonymity in the strongest case. Then, Laguillaumie *et al.* [20] proposed the first scheme with the signature size logarithmic in N, at the cost of relatively large parameters. Later, simpler and more efficient solutions to the signature size were subsequently provided by Nguyen *et al.* [21] and Ling *et al.* [22]. Libert *et al.* [23] obtained substantial efficiency improvements via a construction based on merkle trees that eliminate the need for Gentry-Peikert-Vaikuntanathan trapdoors [24]. In 2016, a scheme supporting message-dependent opening [25] was proposed in [26]. All the schemes mentioned above were designed for static groups, and all have signature sizes that are dependent on N. Three LGSS were proposed by Langlois *et al.* [27], Libert *et al* [28], and Ling *et al.* [22], which have certain dynamic features. Recently, Ling *et al.* [9] constructed the first constant-size group signature from lattices, and the scheme is based on the DMS. Katsumata *et al.* [29] made group signatures without NIZK from lattices in the standard model.

B. OUR CONTRIBUTIONS

In this paper, we propose a new constant-size group signature scheme from lattices. First, we use a variant of the trapdoor generation algorithm to enroll new users. This approach reduces the number of public matrices to two, which reduces the size of both the public key and private key. Second, the scheme follows the sign hybrid encrypt protocol; at the core of its design is the LSS. We make a double Lyubashevsky signature. The LSS can promote the efficiency of the scheme and reduce the signature size. By contrast, the LSS ensures that when the input of the open algorithm is a valid signature signed by a real signer, it is impossible to output another signer's identity. Additionally, the result of the encryption scheme is used as some of the input for generating the random vector in the LSS, and some results of the LSS are used as the plaintext of the encryption scheme (sign-hybrid-encrypt). Moreover, our scheme is anonymous, in addition to traceable in the random oracle model. Furthermore, the security of the scheme is based on the Ring Learning With Errors (RLWE) and Ring Short Integer Solution (RSIS) assumptions that provide optimal performance for the scheme.

The remainder of the paper is organized as follows: In Section 2, we provide the basic notation and some preliminary information. In Section 3, we recall the definition and security model of the group signature. In Section 4, we present our scheme. In Section 5, we analyze the scheme. Finally, we conclude the paper in Section 6.

II. PRELIMINARIES

A. BASIC NOTATION

Let $k \in \mathbb{N}$ be a positive integer and $n = 2^k$, q be a positive prime such that $q \equiv 1(\mod 2n)$. Consider rings of the form $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$, where $\mathbb{Z}[x]$ is the polynomial ring with coefficients in $\mathbb{Z}$ and $x^n + 1$ is the cyclotomic polynomial of degree n. Represent this set using coefficients in the range $[-(q-1)/2, (q-1)/2]$.

For any probability distribution f over $\mathbb{Z}_q$, if $\boldsymbol{v} \in \mathcal{R}_q$, then $\boldsymbol{v} \leftarrow f^n$ indicates that each coefficient of $\boldsymbol{v}$ is chosen from f. If $g : Z \rightarrow \mathbb{R}$ is a probability distribution, then $z \leftarrow g$ denotes element z that is chosen from Z according to probability distribution g.

We write vector V as $\boldsymbol{V} = (v_1, \cdots, v_n)^T$. The first norm is $\|\boldsymbol{V}\|_1 = \sum_{i=1}^{n} |v_i|$, the Euclidean norm is $\|\boldsymbol{V}\| = \|\boldsymbol{V}\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$, and the infinity norm is $\|\boldsymbol{V}\|_\infty = \max_{1 \le i \le n} |v_i|$. Additionally, we write $\mathcal{B}_{m,\kappa} \triangleq \left\{\boldsymbol{x} \in \{0, 1, -1\}^m \,\middle|\, \|\boldsymbol{x}\|_1 = \kappa\right\}$. If n is an integer, then $[n] = \{1, \cdots, n\}$.

B. GAUSSIAN DISTRIBUTION

We next recall the discrete Gaussian distribution, which is a common distribution on lattices. Before presenting the discrete Gaussian distribution, we introduce the continuous Gaussian distribution. The continuous Gaussian distribution over $\mathbb{R}^n$ centered on $\mathbf{v}\in \mathbb{R}^n$ with standard deviation σ is determined by its probability density function

$$\rho^n_{\boldsymbol{v},\sigma}(\boldsymbol{x}) \triangleq \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \cdot \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{v}\|_2^2}{2\sigma^2}\right).$$

Subscript $\boldsymbol{v}$ is usually omitted when $\boldsymbol{v} = \mathbf{0} \in \mathbb{R}^n$. Likewise, discrete Gaussian distribution $\mathcal{D}^n$ over $\mathbb{Z}^n$ centered at some $\boldsymbol{v} \in \mathbb{Z}^n$ with a standard deviation is determined by its probability mass function

$$\mathcal{D}^n_{\boldsymbol{v},\sigma}(\boldsymbol{x}) \triangleq \frac{\rho^n_\sigma(\boldsymbol{x})}{\rho^n_\sigma(\mathbb{Z}^n)}.$$

Additionally, subscript $\boldsymbol{v}$ is also usually omitted when $\boldsymbol{v} = \mathbf{0} \in \mathbb{Z}^n$.

Then, we recall two bounds for the discrete Gaussian distribution that is used in our scheme.

Lemma 1 (Discrete-Gaussian-Bound): ([13]) For any vector $\boldsymbol{v} \in \mathbb{R}^n$ and $\sigma, r > 0$, we have

$$\Pr_{z \leftarrow \mathcal{D}^n_\sigma}[|\langle z, \boldsymbol{v}\rangle| > r] \le 2 \cdot \exp\left(-\frac{r^2}{2\|\boldsymbol{v}\|^2\sigma^2}\right).$$

In particular, for fixed $\kappa > 0$, when $\zeta > \left\lceil \sqrt{\frac{2+2\kappa}{\log_2 e}} \right\rceil$, we have

$$\Pr_{z \leftarrow \mathcal{D}_\sigma^n} \left[|\langle \boldsymbol{z}, \boldsymbol{v} \rangle| > \zeta \|\boldsymbol{v}\| \sigma \right] \le 2^{-\kappa}.$$

Lemma 2 (Bound-for-M): ([13]) For any $\boldsymbol{v} \in \mathbb{Z}^n$, if $\sigma = \omega(\|\boldsymbol{v}\| \cdot \sqrt{n})$, then

$$\Pr_{z \leftarrow \mathcal{D}_\sigma^n} \left[\frac{\mathcal{D}_\sigma^n(\boldsymbol{z})}{\mathcal{D}_{\boldsymbol{v},\sigma}^n(\boldsymbol{z})} = O(1) \right] = 1 - 2^{-\omega(\log n)}.$$

In particular, for any $\boldsymbol{v} \in \mathbb{Z}^n$, if $\zeta > \left\lceil \sqrt{\frac{2+2\kappa}{\log_2 e}} \right\rceil$ and $\sigma = t \cdot \|\boldsymbol{v}\|$ for some $t > 0$, then

$$\Pr_{z \leftarrow \mathcal{D}_\sigma^n} \left[\frac{\mathcal{D}_\sigma^n(\boldsymbol{z})}{\mathcal{D}_{\boldsymbol{v},\sigma}^n(\boldsymbol{z})} < \exp\left(\frac{\zeta}{t} + \frac{1}{2t^2} \right) \right] > 1 - 2^{-\kappa}.$$

In the remainder of the paper, $\mathcal{D}_\sigma$ is defined over $\mathcal{R}_q$, which is similar to the above definition.

C. RLWE PROBLEM AND RSIS PROBLEM

We now recall the RLWE and RSIS problems whose hardness ensures the security of our work. The RLWE problem was introduced by Lyubashevsky *et al.* [33] together with a worst-case to average-case reduction to certain "hard" problems over ideal lattices.

Definition 1 (RLWE Distribution): RLWE distribution $D_{\boldsymbol{s},\chi}$, indexed by $\boldsymbol{s} \in \mathcal{R}_q$ and some pre-defined distribution χ on $\mathcal{R}_q$, is defined as the distribution that samples $\boldsymbol{a} \leftarrow \mathcal{R}_q, \boldsymbol{e} \leftarrow \chi$ and then outputs $(\boldsymbol{a}, \boldsymbol{as} + \boldsymbol{e}) \in \mathcal{R}_q \times \mathcal{R}_q$.

Definition 2 (RLWE Problem): The decision $\mathrm{RLWE}_{n,q,\chi}$ problem is to distinguish, with non negligible advantage, between any desired number of independent samples drawn from $D_{\boldsymbol{s},\chi}$ for a single $\boldsymbol{s} \leftarrow \mathcal{R}_q$, and the same number of uniformly random and independent samples over $\mathcal{R}_q \times \mathcal{R}_q$.

The RSIS problem was proposed by Alwen and Peikert. The average-case RSIS problem is at least as hard as the SVP using a polynomial time quantum reduction. More details are available in [30].

Definition 3 (RSIS Problem): When the parameters n,m,q, β are given, for a uniformly random $\boldsymbol{A} \leftarrow (\mathcal{R}_q)^{1\times m}$, the problem is to determine a non-zero $\boldsymbol{v} \in (\mathcal{R}_q)^m$ such that $\boldsymbol{Av} = \boldsymbol{0} \in \mathcal{R}_q$ and $\|\boldsymbol{v}\|_\infty \le \beta$.

D. TRAPDOOR GENERATION ALGORITHM

We recall the trapdoor generation algorithm, which is used to enroll new users in our construction after some transformation.

Lemma 3 [12],[31],[32]: Let $n, m, q > 0$ be integers, where q is prime. A polynomial time algorithm exists:

- $\mathrm{TrapGen}(n, m, q) \to (\boldsymbol{A}, \boldsymbol{T_A})$: a randomized algorithm that when $m = \Theta(n \times \log q)$, outputs full rank matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and basis $\boldsymbol{T_A}$ of $\Lambda_q^\perp(A)$ (which implies, in particular $\boldsymbol{A} \cdot \boldsymbol{T_A} = 0 \mod q$) such that $\boldsymbol{A}$ is negl(n)-close to uniform and $\|\boldsymbol{T_A}\|_{GS} = \mathcal{O}(\sqrt{n \log q})$, with all but negligible probability in n,

where $\Lambda_q^\perp(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m; \boldsymbol{Ay} = 0 \mod q\}$, $\|\boldsymbol{T}\|_{GS} = \|\boldsymbol{T'}\|$ ($\boldsymbol{T'}$ is the Gram-Schmidt orthogonalization of $\boldsymbol{T}$).

Note that for any $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, we can view $\boldsymbol{A}$ as an m-vector of $\mathcal{R}_q$ according to the following method: Divide $\boldsymbol{A}$ into m blocks, that is, $\boldsymbol{A} = (\boldsymbol{a}_1, \cdots, \boldsymbol{a}_m)$. Then $\boldsymbol{a}_i$ has n elements, that is, $\boldsymbol{a}_i = (a_{i,0}, \cdots, a_{i,n-1})^{\mathrm{T}}$ for any $i \in [m]$, which can be the same as coefficients of an element in $\mathcal{R}_q$. The method is also equal to $(1, \cdots, x^{n-1})\boldsymbol{A}$, that is, an m-vector of $\mathcal{R}_q$. Additionally, for any m-vector of $\mathcal{R}_q$, there is a corresponding element that is constructed by the coefficients of the m-vector. Clearly, the corresponding element is in $\mathbb{Z}_q^{n\times m}$. Hence, *Lemma 3* can be used on $\mathcal{R}_q$. For simplicity, we write $(\boldsymbol{A}, \boldsymbol{T_A}) \leftarrow \mathrm{TrapGen}_{\mathcal{R}_q}(n, m, q)$, where $\boldsymbol{A} \in \mathcal{R}_q^{1\times m}$ and $\boldsymbol{T_A} \in \mathbb{Z}_q^{m\times m}$.

Lemma 4: [32] Let $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{T}_A \in \mathbb{Z}_q^{m\times m}$ be a basis of $\Lambda_q^\perp(\boldsymbol{A})$. Let $\boldsymbol{U} \in \mathbb{Z}_q^{n\times k}$. There is a polynomial time algorithm that outputs $\boldsymbol{X} \in \mathbb{Z}_q^{m\times k}$ that satisfies $\boldsymbol{AX} = \boldsymbol{U} \mod q$ with the following property: $\mathrm{SampleD}(\boldsymbol{A}, \boldsymbol{T_A}, \boldsymbol{U}, \sigma) \to (\boldsymbol{X})$: a randomized algorithm that when $\sigma = \|\boldsymbol{T_A}\|_{GS} \cdot \omega(\sqrt{\log m})$, outputs random sample $\boldsymbol{X}$ from a distribution that is statistically close to $\mathcal{D}_\sigma(\Lambda_q^\perp(\boldsymbol{A}))$.

In particular, if $k = n$, then

$$(1, \cdots, x^{n-1})\boldsymbol{AX}(1, \cdots, x^{n-1})^{\mathrm{T}} = (1, \cdots, x^{n-1})\boldsymbol{U}(1, \cdots, x^{n-1})^{\mathrm{T}}.$$

Suppose that $\boldsymbol{U} = u_1 + u_2 x + \cdots + u_n x^{n-1}$ is an element of $\mathcal{R}_q$, then $u_i \in \mathbb{Z}_q$ for all $i \in [n]$. $(u_1, \cdots, u_n)^{\mathrm{T}}$ can be viewed as the first row of an $n \times n$ matrix $\boldsymbol{U} \in \mathbb{Z}_q^{n\times n}$ where the remaining rows are zero. Then we obtain a similar result over $\mathcal{R}_q$ by *Lemma 4*. For simplicity, we write $\boldsymbol{X} \leftarrow \mathrm{SampleD}_{\mathcal{R}_q}(\boldsymbol{A}, \boldsymbol{T_A}, \boldsymbol{U}, \sigma)$, where $\boldsymbol{A} \in \mathcal{R}_q^{1\times m}$, $\boldsymbol{T_A} \in \mathbb{Z}_q^{m\times m}$, $\boldsymbol{U} \in \mathcal{R}_q$ and $\boldsymbol{X} \in \mathcal{R}_q^{m\times 1}$.

E. REJECTION SAMPLING

We next recall rejection sampling, which is a useful technique to transform an arbitrary distribution to the desired distribution. In many signature schemes, we always store the information about the private key from the output distributions using rejection sampling.

Lemma 5 [13]: Let $f : \mathbb{Z}^n \to \mathbb{R}$ be a probability distribution. Given subset $V \subseteq \mathbb{Z}^n$, let $h : V \to \mathbb{R}$ be a probability distribution defined on V. Let $g_{\boldsymbol{v}} : \mathbb{Z}^n \to \mathbb{R}$ be a family of probability distributions indexed by $\boldsymbol{v} \in V$ such that for almost all $\boldsymbol{v}$'s from h, there exists universal upper bound $M \in \mathbb{R}$ such that

$$\Pr_{z \leftarrow f}[M \cdot g_{\boldsymbol{v}}(z) < f(z)] = \varepsilon.$$

Then the output distribution of the following two algorithms has a negligible statistical difference:

```
$\boldsymbol{v} \leftarrow h$;
$z \leftarrow g_{\boldsymbol{v}}$;
Output $(z, \boldsymbol{v})$ with probability $\min\left(1, \frac{f(z)}{M \cdot g_{\boldsymbol{v}}(z)}\right)$;
```

```
$\boldsymbol{v} \leftarrow h$;
$z \leftarrow f$;
Output $(z, \boldsymbol{v})$ with probability $\frac{1}{M}$.
```

The statistical distance between the output of the above two algorithms is $\frac{\varepsilon}{M}$.

F. LYUBASHEVSKY SIGNATURE SCHEME

We consider the LSS, in which the first signature is based on SIS without a trapdoor. The scheme does not use the hash-and-sign model and uses rejection sampling to obtain the desired distribution. Simultaneously, the public key, private key, and signature size are smaller than those in previous studies. Then, we briefly present the scheme over $\mathbb{Z}_q$.

Key generation:

- signing key $\boldsymbol{S} \leftarrow \{-d, \cdots, d\}^{m\times k}$
- verification key $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\boldsymbol{T} = \boldsymbol{AS}$

Signing: $(\nu, \boldsymbol{A}, \boldsymbol{S})$

- $\boldsymbol{y} \leftarrow \mathcal{D}_\sigma^m$
- $\boldsymbol{c} = H(\boldsymbol{Ay}, \mu)$
- $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{Sc}$ with a probability of min $\frac{\mathcal{D}_z^m(z)}{M\mathcal{D}_{y,Sc}^m(z)}$

Verification:$(\mu, \boldsymbol{z}, \boldsymbol{c}, \boldsymbol{A}, \boldsymbol{T})$

- Accept iff $\|z\|_\infty \leq \eta\sqrt{m} \wedge \boldsymbol{c} = H(\boldsymbol{Az} - \boldsymbol{Tc}, \mu)$, where η is determined by d and σ.

Later, Lyubashevsky proposed a variant of the scheme based on the RSIS assumption. The details are available in [34].

III. GROUP SIGNATURE, ANONYMITY, AND TRACEABILITY

In this section, we recall the definition of group signature by Bellare *et al.* [35].

Definition 4 (Group Signature): A group signature scheme $\prod$ consists of four polynomial-time algorithms (GKg,GSig, GVer,Open):

- GKg(1^λ,1^N): a randomized group key generation algorithm that inputs 1^λ and 1^N, where λ is the security parameter and N is the group size, and returns group public key gpk, group manager's secret key $gmsk$ and player's secret signing key gsk.
- GSig(gpk, $gsk[i]$, μ): a randomized group signing algorithm that inputs group public key gpk, user's key $gsk[i]$ and message μ, and returns signature Σ of μ under $gsk[i]$.
- GVer(gpk, μ, Σ): a deterministic group signature verification algorithm that inputs group public key gpk, message μ and signature Σ, and returns *Accept* or *Reject*.
- Open(gpk, $gmsk$, μ, Σ): a deterministic opening algorithm that inputs group public key gpk, group manager's secret key $gmsk$, message μ and signature Σ, and returns index $i \in [N]$ or $\perp$.

Correctness: Scheme $\prod$ is correct if it satisfies the following requirements: For all $\lambda, N \in \mathbb{N}$, all $(gpk, gmsk, gsk) \leftarrow$GKg $(1^\lambda, 1^N)$, all $i \in [N]$, and all $\mu \in \{0, 1\}^*$,

- verification correctness:
  GVer(gpk, μ, GSig(gpk, $gsk[i]$, μ)) = *Accept*
- opening correctness:
  Open(gpk, $gmsk$, μ, GSig(gpk, $gsk[i]$, μ)) = i

The security requirements for the group signature have two aspects: anonymity and traceability. Before defining them, we introduce oracles that may be used by adversaries in security games.

- Signing oracle $\mathcal{SO}(gpk, gsk[\cdot], \cdot)$: inputs user's index $j \in [N]$ and message μ, and returns valid signature Σ of j for μ.
- Opening oracle $\mathcal{OO}(gpk, gmsk, \cdot, \cdot)$: inputs message μ and signature Σ. If Σ is generated by user $j \in [N]$ for μ, then returns the identity of user j; otherwise, returns $\perp$.
- Corrupt oracle $\mathcal{CO}(\cdot)$: inputs user's index $j \in [N]$ and outputs corresponding secret key $gsk[j]$.

Definition 5 (Full Anonymity): Group signature scheme $\prod$ is anonymous if for any PPT adversary $\mathcal{A}$ and any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the following game is negligible:

- Challenger $\mathcal{C}$ runs the group key generation algorithm with security parameter λ and group size N, and generates keys gpk, $gmsk$ and gsk. Then $\mathcal{C}$ sends gpk and gsk to adversary $\mathcal{A}$.
- Adversary $\mathcal{A}$ is given access to opening oracle $\mathcal{OO}(gpk, gmsk, \cdot, \cdot)$.
- $\mathcal{A}$ provides message μ and two valid identities $1 \leq i_0, i_1 \leq N$. $\mathcal{C}$ randomly selects $b \in \{0, 1\}$ and produces signature $\Sigma^* =$ GSig(gpk, $gsk[i_b]$, μ). Then Σ^* is sent to $\mathcal{A}$.
- $\mathcal{A}$ outputs guess $b' \in \{0, 1\}$, and requires the following conditions:
  - $b' = b$.
  - $\mathcal{A}$ did not query opening oracle $\mathcal{OO}(gpk, gmsk, \cdot, \cdot)$ with μ and Σ^*.

Definition 6 (Full Traceability): Group signature scheme $\prod$ is traceable if for any PPT adversary $\mathcal{A}$ and any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the following game is negligible:

- Challenger $\mathcal{C}$ runs the group key generation algorithm with security parameter λ and group size N, and generates keys gpk, $gmsk$ and gsk. Then $\mathcal{C}$ sends gpk and $gmsk$ to the adversary $\mathcal{A}$.
- Adversary $\mathcal{A}$ is given access to signing oracle $\mathcal{SO}(gpk, gsk[\cdot], \cdot)$ and corrupt oracle $\mathcal{CO}(\cdot)$.
- Adversary $\mathcal{A}$ outputs forgery (μ^*, Σ^*), and requires the following conditions:
  - Ver(gpk, μ^*, Σ^*) = *Accept*.
  - One of the following two conditions is satisfied:
    - $*$ Open(gpk, $gmsk$, μ^*, Σ^*) = $\perp$.
    - $*$ $\exists j^* \in [N]$ such that Open(gpk, $gmsk$, μ^*, Σ^*) $= j^* \wedge ((j^*, \mu^*)$ and j^* not queried by $\mathcal{A}$).

IV. GROUP SIGNATURE SCHEME

In this section, we describe the constant-size group signature scheme based on the lattice via the following four algorithms, where H_1 and H_2 are two different hash functions to $\mathcal{B}_{m,\kappa}$.

Algorithm 1 Verification Algorithm

```
Input: $(gpk, \mu, \Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2))$
Output: Accept or Reject
$\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_3)$
$\boldsymbol{w}_2 = \boldsymbol{B}z_1 + \boldsymbol{A}z_2 - \boldsymbol{u}\boldsymbol{c} \mod q$
$\boldsymbol{c'}_2 = H_1(\mu, \boldsymbol{w}_2, \boldsymbol{t}_1, \boldsymbol{t}_2)$
if $(\boldsymbol{c}_2 = \boldsymbol{c'}_2) \wedge (\|z_1\|_\infty \leq B) \wedge (\|z_2\|_\infty \leq B)$ then
    return "Accept"
else
    return "Reject"
end if
```

A. GROUP KEY GENERATION ALGORITHM

In **Algorithm 2**, where λ is the security parameter and N is the number of group members, assume that $\|\boldsymbol{T_A}\|_{GS} = L$ and $l = \lfloor \log(q-1)/2 \rfloor + 1$. Then, the group manager creates a group public key that consists of two parts: (i) verification key $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{u})$ to the LSS, where $\boldsymbol{B}$ is also used for users to generate their short secret vectors with public syndromes as user key pairs; and (ii) two public keys from an extended version of the LPR encryption scheme [33]. The open key is the corresponding secret key of the two public keys.

Algorithm 2 Key Generation Algorithm

```
1: Input: $(1^\lambda, 1^N)$
2: Output: $(gpk, gmsk, gsk)$
3: $(\boldsymbol{A}, \boldsymbol{T_A}) \leftarrow$TrapGen$_{\mathcal{R}_q}(n, m, q)$
4: $\boldsymbol{a} \leftarrow \mathcal{R}_q^{1\times l}$
5: $\boldsymbol{s}, \boldsymbol{e} \leftarrow \mathcal{D}^l_{\sigma_1}$
6: $\boldsymbol{b} = \boldsymbol{a}\boldsymbol{s} + \boldsymbol{e} \mod q$
7: $\boldsymbol{u} \leftarrow \mathcal{R}_q$
8: $\boldsymbol{B} \leftarrow \mathcal{R}_q^{1\times m}$
9: for all $i$ such that $1 \leq i \leq N$ do
10:     $\boldsymbol{x}_{i1} \leftarrow \mathcal{D}^m_{\sigma_2}$
11:     if $\sum_{k=1}^{\kappa} max_k(\boldsymbol{x}_{i1}) > U$, then go to step 10 and restart
12:     $\boldsymbol{g}_i = \boldsymbol{B}\boldsymbol{x}_{i1} \mod q$
13:     $\boldsymbol{x}_{i2} \leftarrow$SamlpeD$_{\mathcal{R}_q}(\boldsymbol{A}, \boldsymbol{T_A}, \boldsymbol{u} - \boldsymbol{g}_i, \sigma_2)$
14:     if $\sum_{k=1}^{\kappa} max_k(\boldsymbol{x}_{i2}) > U$, then go to step 13 and restart
15:     $gsk[i] = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2})$
16: end for
17: outputs $gpk = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{u}, \{\boldsymbol{g}_i\}_{i=1}^N)$, $gmsk = \boldsymbol{s}$ and $gsk = \{gsk[i]\}_{i=1}^N$
```

To enroll new users, for all $i \in [N]$, the group manager runs SampleD$_{\mathcal{R}_q}(\boldsymbol{A}, \boldsymbol{T_A}, \boldsymbol{u} - \boldsymbol{g}_i, \sigma_2)$ (as in Section 2) to obtain $\boldsymbol{x}_{i2} \in \mathcal{D}^{m\times n}_{\sigma_2}$ such that $\boldsymbol{A}\boldsymbol{x}_{i2} + \boldsymbol{g}_i = \boldsymbol{u}$, where $\boldsymbol{g}_i$ is the user's public key.

In the generation of the user's identity and the process of user joining the group, $\sum_{k=1}^{\kappa} max_k(\cdot)$ is an operator for generating the κ largest entries of the input. We require that the κ largest entries of $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ are both smaller than U to obtain the desired distribution of z_1 and z_2 in the signing algorithm.

Hence, considering the public key of users, the group public key is $gpk = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{u}, \boldsymbol{a}, \boldsymbol{b}, \{\boldsymbol{g}_i\}_{i=1}^N)$, the group manager's secret key is $gmsk = \boldsymbol{s}$ and the user's secret key is $gsk = \{gsk[i]\}_{i=1}^N$, where $gsk[i] = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2})$.

Remark 1: In the group key generation algorithm, it is impossible for $\boldsymbol{e}$ to be public. In the intervening time, $\boldsymbol{e}$ is not used in the group signing algorithm. Hence, it is also unnecessary to store $\boldsymbol{e}$ as the private key.

Algorithm 3 Signing Algorithm

```
Input: $(gpk, gsk_\pi, \mu)$
Output: $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2)$
$\boldsymbol{s}_1, \boldsymbol{e}_1, \boldsymbol{e}_2 \leftarrow \mathcal{D}^l_{\sigma_1}$
$\boldsymbol{t}_1 = \boldsymbol{a}\boldsymbol{s}_1 + \boldsymbol{e}_1 \mod q$
$\boldsymbol{t}_2 = \boldsymbol{b}\boldsymbol{s}_1 + \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rfloor \boldsymbol{g'}_\pi$
$\boldsymbol{y}_1, \boldsymbol{y}_2 \leftarrow [-B, B]^m$
$\boldsymbol{v}_1 = \boldsymbol{B}\boldsymbol{y}_1 \mod q$
$\boldsymbol{v}_2 = \boldsymbol{B}\boldsymbol{y}_1 + \boldsymbol{A}\boldsymbol{y}_2 \mod q$
$\boldsymbol{c}_1 = H_1(\mu, \boldsymbol{v}_1, \boldsymbol{t}_2, \boldsymbol{t}_1)$
$\boldsymbol{c}_2 = H_1(\mu, \boldsymbol{v}_2, \boldsymbol{t}_1, \boldsymbol{t}_2)$
$\boldsymbol{t}_3 = \boldsymbol{b}\boldsymbol{s}_1 - \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rfloor \boldsymbol{c'}_1$
$\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_3)$
$z_1 = \boldsymbol{y}_1 + \boldsymbol{x}_{\pi 1}\boldsymbol{c}$
Repeat with probability $1 - \min\left(1, \frac{\mathcal{D}^m_z(z)}{M\cdot\mathcal{D}^m_{y,SC}(z)}\right)$
$z_2 = \boldsymbol{y}_2 + \boldsymbol{x}_{\pi 2}\boldsymbol{c}$
Repeat with probability $1 - \min\left(1, \frac{\mathcal{D}^m_z(z)}{M\cdot\mathcal{D}^m_{y,SC}(z)}\right)$
output $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2)$
```

B. GROUP SIGNING ALGORITHM

In **Algorithm 3**, to obtain a signature for message μ, user π first encrypts binary representation $\boldsymbol{g'}_\pi$ of user's public key $\boldsymbol{g}_\pi$ using the two public keys. Then, user π randomly selects two vectors from $[-B, B]^m$, that is, a uniform distribution over $\mathcal{R}_q$, and generates two hash values by combining values from two vectors and two public matrices with the above ciphertexts and message μ. Later, user π encrypts the binary representation $\boldsymbol{c'}_1$ of the hash value $\boldsymbol{c}_1$ that can be verified by the user's public key directly. Finally, user π makes a hash function on the other hash value $\boldsymbol{c}_2$ and the ciphertext, and generates the final result of the LSS using rejection sampling. The signature contains all ciphertexts, hash value $\boldsymbol{c}_2$, and two results of the LSS except the hash value.

The concept of the design essentially borrows the Lyubashevshky signature model. Additionally, we introduce an IND-CCA encryption scheme to hide the signer's identity. In fact, we encrypt each coefficient of $\boldsymbol{g}_\pi$ and $\boldsymbol{c}_1$ using the IND-CCA encryption scheme. Simultaneously, for anonymity and traceability, we construct another ciphertext $\boldsymbol{t}_3$ that contains an intermediate product of the LSS. In this way, the signing algorithm is very effective, and the signature size is much shorter than that in previous studies.

C. GROUP VERIFICATION ALGORITHM

In **Algorithm 1**, the group verification algorithm is similar to the LSS [13].

Algorithm 4 Opening Algorithm

```
Input: $(gpk, gmsk, \mu, \Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2))$
Output: $i \in [N]$ or $\perp$
$\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_3)$
$\boldsymbol{g''} = (g_i) = \boldsymbol{t}_2 - \boldsymbol{t}_1\boldsymbol{s}$ where $i = 1, \cdots, nl$
for all $i$ such that $1 \le i \le nl$ do
    if $g_i \approx \lfloor \frac{q}{2} \rfloor$ then
        $g'_i = 1$
    else
        $g'_i = 0$
    end if
end for
for all $j$ such that $1 \le j \le n$ do
    $g''_j = \sum_{i=1}^{l} 2^{i-1} g'_{i+j}$
end for
$\boldsymbol{g}_k = (g''_j)$
$\boldsymbol{c''}_1 = (c_i) = \boldsymbol{t}_3 - \boldsymbol{t}_1\boldsymbol{s}$ where $i = 1, \cdots, nl$
for all $i$ such that $1 \le i \le nl$ do
    if $c_i \approx \lfloor \frac{q}{2} \rfloor$ then
        $c'_i = 1$
    else
        $c'_i = 0$
    end if
end for
for all $j$ such that $1 \le j \le n$ do
    $c'_j = \sum_{i=1}^{l} 2^{i-1} c'_{i+j}$
    $\boldsymbol{c}_1 = (c'_j)$
end for
if $\boldsymbol{c}_1 = H_1(\mu, \boldsymbol{B}z_1 - \boldsymbol{g}_k\boldsymbol{c} \mod q, \boldsymbol{t}_2, \boldsymbol{t}_1) \wedge \boldsymbol{g}_k \in gpk$ then
    return $k$
else
    return $\perp$
end if
```

D. GROUP OPENING ALGORITHM

In **Algorithm 4**, after inputting group public key gpk, group manager's secret key $gmsk$, message μ and signature $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2))$, the group manager computes $\boldsymbol{t}_2 - \boldsymbol{t}_1\boldsymbol{s}$, $\boldsymbol{t}_3 - \boldsymbol{t}_1\boldsymbol{s}$, and $\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_3)$. If the bound of the entry of $\boldsymbol{t}_2 - \boldsymbol{t}_1\boldsymbol{s}$ is smaller than $d(\ll q)$ (the bound of $\boldsymbol{e}_2 + \boldsymbol{e}\boldsymbol{s}_1 - \boldsymbol{e}_1\boldsymbol{s}$), then the corresponding entry g' of $\boldsymbol{g'}_\pi$ is zero. If the bound of the entry of $\boldsymbol{t}_2 - \boldsymbol{t}_1\boldsymbol{s}$ is close to $\lfloor \frac{q}{2} \rfloor$, then the corresponding entry g' of $\boldsymbol{g'}_\pi$ is one. Then, the group manager can recover $\boldsymbol{g}_\pi = (g_{\pi j}) = (\sum_{i=0}^{l} 2^i g'_{i+j})$ for $j = 1, \cdots, n$. The same approach can recover $\boldsymbol{c}_1$ by $\boldsymbol{t}_3 - \boldsymbol{t}_1\boldsymbol{s}$. $\boldsymbol{c}_1$ can be restored in the same manner. If $\boldsymbol{c}_1 = H_1(\mu, \boldsymbol{B}z_1 - \boldsymbol{g}_\pi\boldsymbol{c}, \boldsymbol{t}_2, \boldsymbol{t}_1)$ and $\boldsymbol{g}_\pi \in gpk$, then the algorithm returns π, otherwise, it returns $\perp$.

We create $\boldsymbol{t}_3$ is because the signature created by user π cannot be forged by any users who do not hold secret key $\boldsymbol{g}_\pi$. In this manner, the open algorithm not only reveals the signer's identity, but also ensures that the signature is only signed by the signer. It means that the open algorithm is more specific. It is not necessary to add another algorithm for the assessment of the result of the open algorithm and signature.

Group signature scheme $\prod$ can be described as above. Many efficient variants on the scheme exist, such as those based on LWE and SIS with the NTT technique, and those based on RLWE and RSIS with a rounding operation.

V. ANALYSIS OF THE GROUP SIGNATURE SCHEME

In this section, we first consider the parameters in the scheme. Then under the requirements of the parameters, the scheme is correct naturally. Finally, we reduce the security to the RLWE and RSIS problems. Thus, the scheme is anonymous and traceable in the random oracle model.

A. PARAMETERS

The parameters of the program are λ,N,q,m,n,L,σ_1,σ_2, κ,B,U,M,d. It is not necessary to fix N in the setup stage. According to the trapdoor algorithm, q,m,n,L,σ_2 satisfy $m = \Theta(n \times \log q)$, $L = \mathcal{O}(\sqrt{n \log q})$ and $\sigma_2 = L \cdot \omega(\sqrt{\log m})$.

Note that $\boldsymbol{y}$ is sampled from a uniform distribution on $[-B, B]^m$, which means that there are $(2B+1)^{2m}$ choices of $\boldsymbol{y}$ and $2B < q$. Additionally, $\boldsymbol{A'y}$ has at most $(2B+1)^m$ choices, where $\boldsymbol{A'} \in \mathcal{R}_q^{1\times 2m}$. Hash output $H(\mu, \boldsymbol{A'y}, \boldsymbol{c}_\star)$ is uniformly distributed, which requires that sufficient choices of $\boldsymbol{A'y}$ are hashed. Thus, $(2B+1)^{2m} \ge 2^\kappa \mathbf{C}_m^\kappa$, which means that the probability of a collision for the hash function is smaller than $\frac{1}{2^\kappa}$. Hence, the probability that the following simulator algorithm is aborted is at most $\frac{1}{2^\kappa}$.

Then, rejection sampling ensures that the distributions of z_1 and z_2 are similar to the uniform distribution on $[-B+U, B-U]^m$, which requires that $U \ll B$ and $U = \mathcal{O}(\sqrt{\kappa}\sigma_2)$ (*e.g.*, $U = 14\sqrt{\kappa}\sigma_2$). To reduce the repeat time, we demand that $M \approx (1 - \frac{2U}{2(B-U)+1})^m$ with $B \gg U$. More choices regarding B and U are available in [13].

Additionally, successful decryption requires that $d < \mathcal{O}(\sigma_1^2) \ll \frac{q}{2}$ and d is as small as possible. For more choices of d, refer to [29].

Remark 2: We consider the κ largest entries of $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ for the pass rate of the signing algorithm. Because $U = \mathcal{O}(\sqrt{\kappa}\sigma_2)$, it is obvious that almost all values of $\boldsymbol{x}_{i1}\boldsymbol{c}$ and $\boldsymbol{x}_{i2}\boldsymbol{c}$ are included in U. The negligible values that are outside U can be disregarded. Thus, the repetition of key generation cannot be taken into consideration.

B. CORRECTNESS

The correctness of the scheme can be proved as follows: Suppose signature $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2)$ is valid for message μ under group public key $gpk = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{u}, \{\boldsymbol{g}_i\}_{i=1}^N)$ by player $\boldsymbol{g}_\pi$.

1) VERIFICATION CORRECTNESS

first generates $\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_2)$ and $\boldsymbol{w}_2 = \boldsymbol{B}z_1 + \boldsymbol{A}z_2 - \boldsymbol{u}\boldsymbol{c}$, then $\boldsymbol{w}_2 = \boldsymbol{B}\boldsymbol{y}_1 + \boldsymbol{A}\boldsymbol{y}_2 = \boldsymbol{v}_2 \mod q$ because of $z_1 = \boldsymbol{y}_1 + \boldsymbol{x}_{\pi 1}\boldsymbol{c}$, $z_2 = \boldsymbol{y}_2 + \boldsymbol{x}_{\pi 2}\boldsymbol{c}$ and $\boldsymbol{A}\boldsymbol{x}_{\pi 2} + \boldsymbol{B}\boldsymbol{x}_{\pi 1} = \boldsymbol{u} \mod q$. Thus, $\boldsymbol{c'}_2 = H_1(\mu, \boldsymbol{w}_2, \boldsymbol{t}_1, \boldsymbol{t}_2) = H_1(\mu, \boldsymbol{v}_2, \boldsymbol{t}_1, \boldsymbol{t}_2) = \boldsymbol{c}_2$.

Additionally, the distributions of z_1 and z_2 are close to a uniform distribution on $[-B, B]^m$ using rejection sampling. Then $(\|z_1\|_\infty \leq B) \wedge (\|z_2\|_\infty \leq B)$. Hence, signature Σ passes the verification.

2) OPENING CORRECTNESS

Because $\boldsymbol{t}_1 = \boldsymbol{a}\boldsymbol{s}_1 + \boldsymbol{e}_1 \mod q, \boldsymbol{t}_2 = \boldsymbol{b}\boldsymbol{s}_1 + \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rfloor \boldsymbol{g'}_\pi$ and $\boldsymbol{t}_3 = \boldsymbol{b}\boldsymbol{s}_1 - \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rfloor \boldsymbol{c'}_1$, where $\boldsymbol{g'}_\pi$ is the binary representation of $\boldsymbol{g}_\pi$. Note that $\boldsymbol{t}_2 - \boldsymbol{t}_1\boldsymbol{s} = \boldsymbol{e}\boldsymbol{s}_1 + \boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{s} + \lfloor \frac{q}{2} \rfloor \boldsymbol{g'}_\pi$, where $\|\boldsymbol{e}\|_\infty \leq U'$, $\|\boldsymbol{e}_2\|_\infty \leq U'$, $\|\boldsymbol{e}_1\|_\infty \leq U'$, $\|\boldsymbol{s}\|_\infty \leq U'$ and $\|\boldsymbol{s}_1\|_\infty \leq U'$, and $U' = \mathcal{O}(\sigma_1)$; that is, $d < 2U'^2 + U' \ll \frac{q}{2}$. The rounding procedure in the open algorithm recovers $\boldsymbol{g}_\pi$ with probability one. The same procedure is easily adapted to recover $\boldsymbol{c}_1$. We observe that $\boldsymbol{v}_1 = \boldsymbol{B}\boldsymbol{y}_1 = \boldsymbol{B}z_1 - \boldsymbol{g}_\pi \boldsymbol{c}$. Then we have $\boldsymbol{c}_1 = H_1(\mu, \boldsymbol{B}z_1 - \boldsymbol{g}_\pi \boldsymbol{c}, \boldsymbol{t}_2, \boldsymbol{t}_1)$. Thus, signature Σ is created by user π.

C. FULL ANONYMITY

Roughly speaking, to prove full anonymity, we first show how we can simulate the signing algorithm by programming the random oracle. Then, we show that an adversary who breaks full anonymity using the simulator algorithm can be used to solve the RLWE problem.

Algorithm 5 The Simulator Algorithm

```
Input: $gpk, \pi, \mu$
Output: $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2)$
$\boldsymbol{s}_1, \boldsymbol{e}_1, \boldsymbol{e}_2 \leftarrow \mathcal{D}^l_{\sigma_1}$
$\boldsymbol{t}_1 = \boldsymbol{a}\boldsymbol{s}_1 + \boldsymbol{e}_1 \mod q$
$\boldsymbol{t}_2 = \boldsymbol{b}\boldsymbol{s}_1 + \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rfloor \boldsymbol{g'}_\pi$
$\boldsymbol{c}_1 \leftarrow \mathcal{B}_{m,\kappa}$
$\boldsymbol{c}_2 \leftarrow \mathcal{B}_{m,\kappa}$
$\boldsymbol{t}_3 = \boldsymbol{b}\boldsymbol{s}_1 - \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rfloor \boldsymbol{c'}_1$
$\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_3)$
$z_1 \leftarrow [-B + U, B - U]^m$
Repeat with probability $1 - \min\left(1, \frac{1}{M}\right)$
$z_2 \leftarrow [-B + U, B - U]^m$
Repeat with probability $1 - \min\left(1, \frac{1}{M}\right)$
$\boldsymbol{w}_2 = \boldsymbol{B}z_1 + \boldsymbol{A}z_2 - \boldsymbol{u}\boldsymbol{c} \mod q$
$\boldsymbol{w}_1 = \boldsymbol{B}z_1 - \boldsymbol{g}_\pi \boldsymbol{c} \mod q$
if $H_1$ has already been defined on $\boldsymbol{w}_1$ or $\boldsymbol{w}_2$ then
    Abort
else
    Program $\boldsymbol{c}_2 = H_1(\mu, \boldsymbol{w}_2, \boldsymbol{t}_1, \boldsymbol{t}_2)$ and $\boldsymbol{c}_1 = H_1(\mu, \boldsymbol{w}_1, \boldsymbol{t}_2, \boldsymbol{t}_1)$
end if
output $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2)$
```

First, we define the simulator algorithm as Algorithm 5. Then we have the following lemma.

Lemma 6: Suppose the parameters are defined as above, and the statistical distance between the output of the signing algorithm and the simulator algorithm, which does not take the user's secret key $gsk[i]$ as input, is at most $\frac{1}{2^{\kappa-1}} + \frac{2^{m+1}U^m}{(2B+1)^m 2M}$, that is, negligible.

Proof: The difference between the signing algorithm and the simulator algorithm is the generation of $z_1, z_2, \boldsymbol{c}_1$, and $\boldsymbol{c}_2$. The value of

$$\begin{aligned} \boldsymbol{c}_2 &= H_1\left(\mu, \boldsymbol{B}\boldsymbol{y}_1 + \boldsymbol{A}\boldsymbol{y}_2, \boldsymbol{t}_1, \boldsymbol{t}_2\right) \\ &= H_1\left(\mu, \boldsymbol{B}z_1 + \boldsymbol{A}z_2 - \boldsymbol{u}\boldsymbol{c}, \boldsymbol{t}_1, \boldsymbol{t}_2\right) \end{aligned}$$

obtains a set uniformly at random in the simulator algorithm, whereas in the signing algorithm, H_1 checks whether H_1 was already evaluated on $(\mu, \boldsymbol{B}\boldsymbol{y}_1 + \boldsymbol{A}\boldsymbol{y}_2, \boldsymbol{t}_1, \boldsymbol{t}_2)$. The simulator algorithm differs from the signing algorithm in the case that the value of $\boldsymbol{B}\boldsymbol{y}_1 + \boldsymbol{A}\boldsymbol{y}_2$ is equal to one of the previous values that was queried. The probability of a collision is at most $\frac{1}{2^\kappa}$. Similar to $\boldsymbol{c}_2$, we obtain the same result for $\boldsymbol{c}_1$. The distribution of z_1 and z_2 in the signing algorithm is indistinguishable from a uniform distribution on $[-B + U, B - U]^m$ using rejection sampling, and the statistical distance between them is at most $2(\frac{(2U)^m}{(2B+1)^m M})$. Thus, the statistical distance between the output of the signing algorithm and the simulator algorithm is at most $\frac{1}{2^{\kappa-1}} + \frac{2^{m+1}U^m}{(2B+1)^m M}$, which is negligible under the parameters set as above.

Theorem 1: Let the parameters be as presented above and κ be sufficiently large. Then the signature scheme is anonymous under the RLWE assumption in the random oracle model.

Proof: We construct a series of games where we make changes to prove the full anonymity of the scheme.

Game 0: Suppose that challenger $\mathcal{C}$ runs the group key generation algorithm with security parameter λ and group size N, and generates keys gpk, $gmsk$ and gsk. Then $\mathcal{C}$ sends gpk and gsk to adversary $\mathcal{A}$. When $\mathcal{A}$ queries $\mathcal{OO}(gpk, gmsk, \cdot, \cdot)$ with (μ', Σ'), index $k \in [N]$ or $\perp$ is returned. Thus, in Game 0, the adversary $\mathcal{A}$ first selects two signers' indexes $i_0, i_1 \in \{1, \cdots, N\}$ with $i_0 \neq i_1$ and message μ, and sends them to challenger $\mathcal{C}$. Then challenger $\mathcal{C}$ chooses $b \in \{0, 1\}$, and calls the real signing algorithm with $(gpk, gsk[i_b], \mu)$. Signature Σ =GSig$(gpk, gsk[i_b], \mu)$ is sent to $\mathcal{A}$, and $\mathcal{A}$ outputs guess b' for signer's index i_b. Thus, in Game 0, $\mathcal{A}$ succeeds in breaking ambiguity Game 0 ($b = b'$) if $\Pr[\text{Game } 0] \leq 1/2 + non - negligible$; otherwise, $\mathcal{A}$ is only randomly guessing.

Game 1: Similar to Game 0, except the real signing algorithm is replaced with the simulator algorithm. From the above lemma, we obtain that $|\Pr[\text{Game } 0] - \Pr[\text{Game } 1]| \leq \frac{1}{2^{\kappa+1}} + \frac{(2U)^m}{(2B+1)^m 2M}$.

In Game 1, challenge signature $\Sigma = (z_1, z_2, \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{c}_2)$ is returned by the simulator algorithm. We observe that the user's identity is only used for generating $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ in the simulator algorithm, which does not have the user's private key. $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ are the only two places where an adversary may obtain some efficient information about the identity of the real signer, and also are the ciphertexts of the following encryption scheme $\mathcal{E}$.

$$\mathrm{Gen}(1^{\lambda}):\left\{\begin{array}{l}\boldsymbol{a}\leftarrow\mathcal{R}_q^{1\times l}\\ \boldsymbol{s},\boldsymbol{e}\leftarrow\mathcal{D}_{\sigma_1}^{l\times n}\\ \boldsymbol{b}=\boldsymbol{as}+\boldsymbol{e}\mod q\\ pk=(\boldsymbol{a},\boldsymbol{b}),sk=\boldsymbol{s}\end{array}\right\}$$

$$\mathrm{Enc}(pk,\boldsymbol{g'}_{\pi},\boldsymbol{c'}_1):\left\{\begin{array}{l}\boldsymbol{s}_1,\boldsymbol{e}_1,\boldsymbol{e}_2\leftarrow\mathcal{D}_{\sigma_1}^{l\times n}\\ \boldsymbol{t}_1=\boldsymbol{as}_1+\boldsymbol{e}_1\mod q\\ \boldsymbol{t}_2=\boldsymbol{bs}_1+\boldsymbol{e}_2+\lfloor\frac{q}{2}\rfloor\boldsymbol{g'}_{\pi}\\ \boldsymbol{t}_3=\boldsymbol{bs}_1-\boldsymbol{e}_2+\lfloor\frac{q}{2}\rfloor\boldsymbol{c'}_1\end{array}\right\}$$

$$\mathrm{Dec}(sk,\boldsymbol{t}_1,\boldsymbol{t}_2,\boldsymbol{t}_3):\left\{\begin{array}{l}\boldsymbol{g'}_{\pi}\approx\boldsymbol{t}_2-\boldsymbol{t}_1\boldsymbol{s}\\ \boldsymbol{c'}_1\approx\boldsymbol{t}_3-\boldsymbol{t}_1\boldsymbol{s}\end{array}\right\}$$

It is easy to check that the probability of Game 1 succeeding is equal to the probability that adversary $\mathcal{A}$ breaks $\mathcal{E}$.

Consider $\mathcal{E}$, whose security game Game 2 can be simplified as follows: Challenger $\mathcal{C}$ first calls key generation algorithm to obtain public keys $(\boldsymbol{a},\boldsymbol{b}=\boldsymbol{as}+\boldsymbol{e}\mod q)$ and secret key $\boldsymbol{s}$, and then sends the public keys to adversary $\mathcal{A}$. Moreover, adversary A can access the encryption oracle and decryption oracle. Thus, $\mathcal{A}$ selects two messages $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ for challenger $\mathcal{C}$, $\mathcal{C}$ calls the encryption algorithm with $\boldsymbol{x}_b$ which is randomly chosen from the two messages, and the plaintext is provided to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs guess b' for encrypted message $\boldsymbol{x}_b$. Note that $\mathcal{E}$ is the extension of LPR encryption scheme [33], which is IND-CCA secure via the Naor-Yung transform [36] under the RLWE assumption. Thus, $\Pr[\text{Game 2}]=1/2+\textit{negligible}$.

Combining the probabilities of the above games, we know that $|\Pr[\text{Game 0}]-\Pr[\text{Game 2}]|\leq\textit{negligible}+\frac{1}{2^{\kappa+1}}+\frac{(2U)^m}{(2B+1)^m2M}$, and then $\Pr[\text{Game 0}]\leq 1/2+\textit{negligible}$. Hence, the signature scheme is anonymous.

D. FULL TRACEABILITY

In this subsection, we prove that the scheme is traceable under the RSIS assumption. Before presenting the theorem, we provide an important lemma, which can be viewed as a corollary of *Lemma* 2 in [9].

Lemma 7: Let $\boldsymbol{B'}\in\mathcal{R}_q^{1\times 2m}$, where $m>\frac{1}{2}\lceil\log q\rceil+\frac{1}{2}$. If $\boldsymbol{x}$ is randomly chosen from $\mathcal{D}^{2m}_{\sqrt{2}\sigma_2}$ such that $\|\boldsymbol{x}\|_{\infty}\leq 7\sigma_2$ ($\sigma_2>1$), then with a probability of at least $1-4^{-n}$, there exists another $\boldsymbol{x'}\leftarrow\mathcal{D}^{2m}_{\sqrt{2}\sigma_2}$ such that $\|\boldsymbol{x'}\|_{\infty}\leq 7\sigma_2$ and $\boldsymbol{B'x}=\boldsymbol{B'x'}\mod q$.

Proof. The proof of the lemma can be found in *Lemma 2* in [9].

Theorem 2: Let the parameters be as presented above and κ, and n are sufficiently large. The signature scheme is traceable under the RSIS assumption in the random oracle.

Proof. Suppose that adversary $\mathcal{A}$ can defeat the traceability of the scheme with non-negligible success probability ε in the game in *Definition 6*. We build a PPT algorithm $\mathcal{B}$ that attacks the RSIS problem with non-negligible probability.

When providing verification key $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{g}_j,\boldsymbol{u})$, simulator $\mathcal{B}$ runs the experiment in Definition 6 faithfully.. When $\mathcal{A}$ queries its signing oracle $\mathcal{SO}$ on $i\in[N]$, message μ and *gpk*: If $i\neq j$, then $\mathcal{B}$ runs the honest signing algorithm and returns GSig(*gpk*, *gsk*[i], μ); If $i=j$, then $\mathcal{B}$ runs the simulator algorithm and returns the output of the simulator algorithm with *gpk*, j and μ. When $\mathcal{A}$ queries its corrupt oracle $\mathcal{CO}$ on $i\in[N]$: If $i\neq j$, then $\mathcal{B}$ returns *gsk*[i]; If $i=j$, then $\mathcal{B}$ returns $\perp$. In particular, regardless of whether index $i\in[N]$ has been queried to corrupt oracle $\mathcal{CO}$ by $\mathcal{A}$, we can replace the real signing algorithm with the simulator algorithm when $\mathcal{A}$ queries the signing oracle with i. Suppose $\mathcal{A}$ makes h hash queries and s signing queries. Finally $\mathcal{A}$ outputs signature $\Sigma=(z_1,z_2,\boldsymbol{t}_1,\boldsymbol{t}_2,\boldsymbol{t}_3,\boldsymbol{c}_2)$ on message μ for the player $j\in[N]$. With non-negligible probability, $\mathcal{A}$ wins the game. Then one of the following two conditons can be satisfied.

The first condition is that Σ is a valid signature with respect to $\boldsymbol{t}_1,\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ for j.

In the random oracle model, with overwhelming probability, $H_1\left(\mu,\boldsymbol{v}_2,\boldsymbol{t}_1,\boldsymbol{t}_2\right)$ and $H_1\left(\mu,\boldsymbol{v}_1,\boldsymbol{t}_2,\boldsymbol{t}_1\right)$ must have been defined. In the following analysis, we simply ignore the collision event with random oracle H_1, which occurs with negligible probability. There exist two possible cases.

The first case is that $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ were returned by the singing oracle when dealing with message μ' made by forger adversary $\mathcal{A}$. Then $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ have been queried. Specifically, $\mathcal{A}$ always obtained a signature of the form $(z'_1,z'_2,\boldsymbol{t}'_1,\boldsymbol{t}'_2,\boldsymbol{t}'_3,\boldsymbol{c}_2)$ on the message μ', where $\boldsymbol{c}'=H_2(\boldsymbol{c}_2,\boldsymbol{t}'_3)$. We define

$$\boldsymbol{v}'_2=\boldsymbol{B}z'_1+\boldsymbol{A}z'_2-\boldsymbol{u}\boldsymbol{c}',\quad \boldsymbol{v}'_1=\boldsymbol{B}z'_1-\boldsymbol{g}_j\boldsymbol{c}'.$$

Then we have

$$\begin{aligned}\boldsymbol{c}_2&=H_1\left(\mu,\boldsymbol{v}_2,\boldsymbol{t}_1,\boldsymbol{t}_2\right)=H_1\left(\mu',\boldsymbol{v}'_2,\boldsymbol{t}'_1,\boldsymbol{t}'_2\right),\\ \boldsymbol{c}_1&=H_1\left(\mu,\boldsymbol{v}_1,\boldsymbol{t}_2,\boldsymbol{t}_1\right)=H_1\left(\mu',\boldsymbol{v}'_1,\boldsymbol{t}'_2,\boldsymbol{t}'_1\right).\end{aligned}$$

In the random oracle model, with overwhelming probability it holds that:

$$\mu=\mu',\quad \boldsymbol{v}_2=\boldsymbol{v}'_2,\ \boldsymbol{v}_1=\boldsymbol{v}'_1,\ \boldsymbol{t}_1=\boldsymbol{t}'_1,\ \boldsymbol{t}_2=\boldsymbol{t}'_2.$$

Additionally, note that $\boldsymbol{t}_3$ and $\boldsymbol{t}'_3$ are the ciphertexts of $\boldsymbol{c}_1$. Then

$$\boldsymbol{t}_3-\boldsymbol{t}_1\boldsymbol{s}=\boldsymbol{t}'_3-\boldsymbol{t}'_1\boldsymbol{s}\Longrightarrow\boldsymbol{t}_3=\boldsymbol{t}'_3\Longrightarrow\boldsymbol{c}=\boldsymbol{c}'.$$

So $(\mu,z_1,z_2,\boldsymbol{t}_1,\boldsymbol{t}_2,\boldsymbol{t}_3,\boldsymbol{c}_2,\boldsymbol{c})\neq(\mu',z'_1,z'_2,\boldsymbol{t}'_1,\boldsymbol{t}'_2,\boldsymbol{t}'_3,\boldsymbol{c}_2,\boldsymbol{c}')$ $=(\mu,z'_1,z'_2,\boldsymbol{t}_1,\boldsymbol{t}_2,\boldsymbol{t}_3,\boldsymbol{c}_2,\boldsymbol{c})$. Then under operator $\mod q$,

$$\boldsymbol{B}(z'_1-z_1)=0,\ \boldsymbol{B}(z'_1-z_1)+\boldsymbol{A}(z'_2-z_2)=0.$$

Because $\|z'_1-z_1\|_{\infty}\leq 2B$ and $\|z'_2-z_2\|_{\infty}\leq 2B$, $(z'_1-z_1,0)$ can be a solution to the $\mathrm{RSIS}_{n,q,m+1,2B}$ problem. By contrast, the valid signature is on message μ for player j queried to the signing oracle, which means that the case does not exist for (j,μ) that was queried by $\mathcal{A}$ according to the full traceability game.

The second case is that

$$\boldsymbol{c}_2=H_1\left(\mu,\boldsymbol{v}_2,\boldsymbol{t}_1,\boldsymbol{t}_2\right),\quad \boldsymbol{c}_1=H_1\left(\mu,\boldsymbol{v}_1,\boldsymbol{t}_2,\boldsymbol{t}_1\right)$$

were not returned by the signing oracle, but obtained from some query to random oracle H_1 on query $(\mu,\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{t}_1,\boldsymbol{t}_2)$. In this case, we rewind the program to the point

$(\mu, \boldsymbol{v}_1, \boldsymbol{c}_1, \boldsymbol{c}_2)$ and $(\mu, \boldsymbol{v}_2, \boldsymbol{t}_2, \boldsymbol{t}_1)$ of defining $H_1\ (\mu, \boldsymbol{v}_2, \boldsymbol{t}_1, \boldsymbol{t}_2)$ and $H_1\ (\mu, \boldsymbol{v}_1, \boldsymbol{t}_2, \boldsymbol{t}_1)$, and redefine the random oracle output such that $\boldsymbol{c}_2' = H_1\ (\mu, \boldsymbol{v}_2, \boldsymbol{t}_1, \boldsymbol{t}_2)$ and $\boldsymbol{c}_1' = H_1\ (\mu, \boldsymbol{v}_1, \boldsymbol{t}_2, \boldsymbol{t}_1)$. With overwhelming probability, $\boldsymbol{c}_1 \neq \boldsymbol{c}_1'$ and $\boldsymbol{c}_2 \neq \boldsymbol{c}_2'$. By the forking lemma, we obtain another valid signature $(\boldsymbol{z}_1', \boldsymbol{z}_2', \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3', \boldsymbol{c}_2')$ on the same message μ with probability $\varepsilon(\frac{\varepsilon}{h+s} - \frac{1}{2^\kappa})$, such that

$$\begin{aligned} \boldsymbol{Bz}_1' - \boldsymbol{g}_j\boldsymbol{c}' &= \boldsymbol{Bz}_1 - \boldsymbol{g}_j\boldsymbol{c} \mod q, \\ \boldsymbol{Bz}_1' + \boldsymbol{Az}_2' - \boldsymbol{uc}' &= \boldsymbol{Bz}_1 + \boldsymbol{Az}_2 - \boldsymbol{uc} \mod q, \end{aligned}$$

where $\boldsymbol{c} = H_2(\boldsymbol{c}_2, \boldsymbol{t}_3) \neq \boldsymbol{c}' = H_2(\boldsymbol{c}_2', \boldsymbol{t}_3')$. Hence

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{z}_1' - \boldsymbol{z}_1) - \boldsymbol{g}_j(\boldsymbol{c}' - \boldsymbol{c}) = 0 \mod q, \\ \boldsymbol{B}(\boldsymbol{z}_1' - \boldsymbol{z}_1) + \boldsymbol{A}(\boldsymbol{z}_2' - \boldsymbol{z}_2) - \boldsymbol{u}(\boldsymbol{c}' - \boldsymbol{c}) = 0 \mod q. \end{aligned}$$

We claim that

$$(\boldsymbol{z}_1' - \boldsymbol{z}_1) - \boldsymbol{x}_{j1}(\boldsymbol{c}' - \boldsymbol{c}) \neq 0, \quad (\boldsymbol{z}_2' - \boldsymbol{z}_2) - \boldsymbol{x}_{j2}(\boldsymbol{c}' - \boldsymbol{c}) \neq 0,$$

where $\boldsymbol{g}_j = \boldsymbol{Bx}_{j1}$ and $\boldsymbol{Ax}_{j2} + \boldsymbol{Bx}_{j1} = \boldsymbol{u}$. {By *Lemma 7*, we know that there is at least a $1 - 4^{-n}$ chance that there exists another key $gsk[j]' = (\boldsymbol{x}_{j1}', \boldsymbol{x}_{j2}')$ such that $\boldsymbol{Bx}_{j1} = \boldsymbol{Bx}_{j1}'$ mod q and $\boldsymbol{Ax}_{j2} = \boldsymbol{Ax}_{j2}'$ mod q. This shows that for $gsk[j]$ with $(\boldsymbol{z}_1' - \boldsymbol{z}_1) - \boldsymbol{x}_{j1}(\boldsymbol{c}' - \boldsymbol{c}) = 0$ and $(\boldsymbol{z}_2' - \boldsymbol{z}_2) - \boldsymbol{x}_{j2}(\boldsymbol{c}' - \boldsymbol{c}) = 0$, there exists $gsk[j]'$ such that $(\boldsymbol{z}_1' - \boldsymbol{z}_1) - \boldsymbol{x}_{j1}'(\boldsymbol{c}' - \boldsymbol{c}) \neq 0$ and $(\boldsymbol{z}_2' - \boldsymbol{z}_2) - \boldsymbol{x}_{j2}'(\boldsymbol{c}' - \boldsymbol{c}) \neq 0$. $\mathcal{A}$ does not know any information about the secret key; hence, we obtain a non-zero answer with a probability of at least $1/2$.}
Because $\|\boldsymbol{z}_1' - \boldsymbol{z}_1\|_\infty \leq 2B$, $\|\boldsymbol{z}_2' - \boldsymbol{z}_2\|_\infty \leq 2B$ and $\|\boldsymbol{c}' - \boldsymbol{c}\|_\infty \leq 2\kappa$, $(\boldsymbol{z}_1' - \boldsymbol{z}_1, \boldsymbol{c}' - \boldsymbol{c})$ can be a solution to the $RSIS_{n,q,m+1,2B}$ problem.

The second condition is that the output of the open algorithm is $\perp$. There exist two possible cases. The first case is that $\boldsymbol{g}_j \notin gpk$, then signature Σ is a valid forgery of the LSS with public keys $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{u})$. The second case is that $\boldsymbol{c}_1 \neq H_1(\mu, \boldsymbol{Bz}_1 - \boldsymbol{g}_j\boldsymbol{c} \mod q, \boldsymbol{t}_2, \boldsymbol{t}_1) \wedge \boldsymbol{g}_j \in gpk$, then signature Σ is a valid forgery of the LSS with public key $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{u})$. Then $\boldsymbol{B}(\boldsymbol{z}_1' - \boldsymbol{z}_1) + \boldsymbol{A}(\boldsymbol{z}_2' - \boldsymbol{z}_2) = 0 \mod q$ or $\boldsymbol{B}(\boldsymbol{z}_1' - \boldsymbol{z}_1) + \boldsymbol{A}(\boldsymbol{z}_2' - \boldsymbol{z}_2) - \boldsymbol{u}(\boldsymbol{c}' - \boldsymbol{c}) = 0 \mod q$. These cases breaks the unforgeability of the LSS that is based on the RSIS assumption, and more details about the proof is similar to the proof of the first condition.

To summarize, the scheme is traceable under the RSIS assumption in the random oracle model.

Remark 3: From the proof of the above theorem, we observe that if either $\boldsymbol{c}_1$ or $\boldsymbol{c}_2$ was queried to the signing oracle, then falsification fails for (j, μ) queried by $\mathcal{A}$. This is the reason that we do not take into account other cases.

E. COMPARISON

In this subsection, we provide a clear size comparison with the work of Ling *el.at* in TABLE 1. Before presenting the table, we provide some notation.

- k is a positive integer, $q = \tilde{\mathcal{O}}(n^4)$
- $l = \lfloor \log(q/2) \rfloor + 1$, $m \geq 2\lceil \log q \rceil + 2$, $\bar{m} = m + k$
- $c > 1$ is a real constant, $d \geq \log_c(\omega(\log n))$, $c_d \geq \lfloor c^d/(c-1) \rfloor$
- $\beta = \tilde{\mathcal{O}}(n)$, $B = \tilde{\mathcal{O}}(n^{5/4})$, $\delta_\beta = \lfloor \log_2 \beta \rfloor + 1$,$\delta_B = \lfloor \log_2 B \rfloor + 1$

TABLE 1. The size comparison with Ling [1].

Term($n \log q$)	Ling[1]	ours
gpk size	$2\bar{m} + m + (d+1)k + 5l + 1$	$2m + 2l + 1$
gmsk size	$l + 1$	l
gsk[·] size	$2\bar{m} + k + 1 + c_d$	$2m$
signature size	$(k + c_dk + \bar{m})\delta_\beta + (l+1)\delta_B + m$	$2m + 3l$

TABLE 2. The size comparison with Ling [1] under the security parameter.

Term	Ling[1]	ours	difference
gpk size	$\mathcal{O}(\lambda \cdot \log^2 \lambda)$	$\mathcal{O}(\lambda \cdot \log^2 \lambda)$	$\mathcal{O}(\lambda \cdot \log^2 \lambda)$
gmsk size	$\mathcal{O}(\lambda \cdot \log \lambda)$	$\mathcal{O}(\lambda \cdot \log \lambda)$	$\mathcal{O}(\lambda \cdot \log \lambda)$
gsk[·] size	$\mathcal{O}(\lambda \cdot \log^2 \lambda)$	$\mathcal{O}(\lambda \cdot \log^2 \lambda)$	$\mathcal{O}(\lambda \cdot \log^2 \lambda)$
signature size	$\mathcal{O}(\lambda \cdot \log^4 \lambda)$	$\mathcal{O}(\lambda \cdot \log^3 \lambda)$	$\mathcal{O}(\lambda \cdot \log^4 \lambda)$

Furthermore, we describe the size difference between the two schemes with the security parameter λ as follows.

VI. CONCLUSION

We presented a new constant-size group signature scheme based on the RLWE and RSIS assumptions. In the key generation algorithm, we used trapdoor algorithms over $\mathcal{R}_q$ to enroll new users and create the common public equation, which is the first difference from [9] and reduces the number of public matrices to two. In the signing algorithm and verification algorithm, the LSS with an IND-CCA encryption scheme constitute the procedure, which is the second difference from [9] and reduce the signature size. In the open algorithm, we not only decrypted the ciphertext from the IND-CCA encryption scheme, but also made a verification from the LSS, which is the third difference from [9] and ensured that the signature was only produced by the signer from the open algorithm. Compared with [9], our construction is efficient in the signing algorithm, more precise on the open algorithm and shorter in the public key, private key, and signature size. However, rejection sampling may cause some loss in efficiency. Moreover, the scheme has full traceability and full anonymity under the RLWE and RSIS assumptions, which allow us to select the high argument. Although, because of the trapdoor in the setup stage, the scheme may restricted in practical applications. To determine efficient group signatures without the trapdoor is an interesting future work. We will consider whether there is another approach to construct the lattice-based constant-size group signature.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their helpful comments and insightful suggestions. The authors are very grateful to Professor Wu for his assistance.

REFERENCES

[1] D. Chaum and E. van Heyst, "Group signatures," in *Proc. 28th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Cologne, Germany: Springer, 1991, pp. 257–265.

[2] J. Camenisch, "Efficient and generalized group signatures," in *Proc. 16th Int. Conf. Theory Appl. Cryptograph. Techn.* Konstanz, Germany: Springer, 1997, pp. 465–479.

[3] L. Chen and T. P. Pedersen, "New group signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Perugia, Italy: Springer, 1994, pp. 171–181.

[4] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Proc. 17th Int. Conf. Cryptol.* Santa Barbara, CA, USA: Springer, 1997, pp. 410–424.

[5] J. Camenisch and M. Michels, "A group signature scheme with improved efficiency," in *Proc. 4th Annu. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Beijing, China: Springer, 1998, pp. 160–174.

[6] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Proc. 20th Int. Conf. Cryptol.* Santa Barbara, CA, USA: Springer, 2000, pp. 255–270.

[7] G. Ateniese and B. de Medeiros, "Efficient group signatures without trapdoors," in *Proc. 9th Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Taipei, Taiwan: Springer, 2003, pp. 246–268.

[8] S. D. Gordon, J. Katz, and V. Vaikuntanathan, "A group signature scheme from lattice assumptions," in *Proc. 16th Annu. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Singapore: Springer, 2010, pp. 395–412.

[9] S. Ling, K. Nguyen, H. Wang, and Y. Xu, "Constant-size group signatures from lattices," in *Proc. 21st Int. Conf. Pract. Theory Public Key Cryptogr.* Rio de Janeiro, Brazil: Springer, 2018, pp. 58–88.

[10] L. Ducas and D. Micciancio, "Improved short lattice signatures in the standard model," in *Proc. 34th Int. Cryptol. Conf.* Santa Barbara, CA, USA: Springer, 2014, pp. 335–352.

[11] M. Bellare, H. Shi, and C. Zhang, "Foundations of group signatures: The case of dynamic groups," in *Proc. Cryptogr. Track RSA Conf.* San Francisco, CA, USA: Springer, 2005, pp. 136–153.

[12] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Proc. 31st Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Cambridge, U.K.: Springer, 2012, pp. 700–718.

[13] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proc. 31st Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Cambridge, U.K.: Springer, 2012, pp. 738–755.

[14] E. Alkim, N. Bindel, J. A. Buchmann, Ö. Dagdelen, E. Eaton, G. Gutoski, J. Krämer, and F. Pawlega, "Revisiting TESLA in the quantum random oracle model," in *Proc. Post-Quantum Cryptogr., 8th Int. Workshop*. Heidelberg, Germany: Springer, 2017, pp. 143–162.

[15] S. Bai and D. Steven Galbraith, "An improved compression technique for signatures based on learning with errors," in *Proc. Cryptogr. Track at RSA Conf.* San Francisco, CA, USA: Springer, 2014, pp. 28–47.

[16] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Proc. 33rd Int. Cryptol. Conf.* Santa Barbara, CA, USA: Springer, 2013, pp. 40–56.

[17] T. Güneysu, V. Lyubashevsky, and T. Pelmann, "Practical lattice-based cryptography: A signature scheme for embedded systems," in *Proc. 14th Int. Workshop*. Leuven, Belgium: Springer, 2012, pp. 530–547.

[18] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS–Dilithium: A lattice-based digital signature scheme," in *Proc. Conf. Cryptograph. Hardw. Embedded Syst.* Amsterdam, The Netherlands: Springer, 2018, pp. 238–268.

[19] J. Camenisch, G. Neven, and M. Rückert, "Fully anonymous attribute tokens from lattices," in *Proc. 8th Conf. Secur. Cryptogr. Netw.* Amalfi, Italy: Springer, 2012, pp. 57–75.

[20] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé, "Lattice-based group signatures with logarithmic signature size," in *Proc. 19th Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Bengaluru, India: Springer, 2013, pp. 41–61.

[21] P. Q. Nguyen, J. Zhang, and Z. Zhang, "Simpler efficient group signatures from lattices," in *Proc. IACR Int. Conf. Pract. Theory Public-Key Cryptogr.* Annapolis, MD, USA: Springer, 2015, pp. 401–426.

[22] S. Ling, K. Nguyen, and H. Wang, "Group signatures from lattices: Simpler, tighter, shorter, ring-based," in *Proc. IACR Int. Conf. Pract. Theory Public-Key Cryptogr.* Annapolis, MD, USA: Springer, 2015, pp. 427–449.

[23] B. Libert, S. Ling, K. Nguyen, and H. Wang, "Zero-knowledge arguments for latticebased accumulators: Logarithmic-size ring signatures and group signatures without trapdoors," in *Proc. 35th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Vienna, Austria: Springer, 2016, pp. 1–31.

[24] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th ACM Symp. Theory Comput.*, Melbourne, VIC, Australia, 2008, pp. 197–206.

[25] Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote, "Group signatures with message-dependent opening," in *Proc. 5th Int. Conf.* Cologne, Germany: Springer, 2012, pp. 270–294.

[26] B. Libert, F. Mouhartem, and K. Nguyen, "A lattice-based group signature scheme with message-dependent opening," in *Proc. 14th Int. Conf. Appl. Cryptogr. Netw. Secur.* London, U.K.: Springer, 2016, pp. 137–155.

[27] A. Langlois, S. Ling, K. Nguyen, and H. Wang, "Lattice-based group signature scheme with verifier-local revocation," in *Proc. 17th Int. Conf. Pract. Theory Public-Key Cryptogr.* Buenos Aires, Argentina: Springer, 2014, pp. 345–361.

[28] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang, "Signature schemes with efficient protocols and dynamic group sigantures from lattice assumptions," in *Proc. 22nd Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Hanoi, Vietnam: Springer, 2016, pp. 373–403.

[29] S. Ling, K. Nguyen, H. Wang, and Y. Xu, "Lattice-based group signatures: Achieving full dynamicity with ease," in *Proc. 15th Int. Conf. Appl. Cryptogr. Netw. Secur.* Kanazawa, Japan: Springer, 2017, pp. 293–312.

[30] S. Katsumata and S. Yamada, "Group signatures without NIZK: From lattices in the standard model," IACR Cryptol. ePrint Arch., Tech. Rep. 2019/221, 2019.

[31] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," in *Proc. 26th Int. Symp. Theor. Aspects Comput. Sci.* Freiburg im Breisgau, Germany: Springer, 2009, pp. 535–553.

[32] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. 26th Int. Colloq.* Prague, Czech Republic: Springer, 1999, pp. 1–9.

[33] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, "Fully key homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *Proc. 3rd Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Copenhagen, Denmark: Springer, 2014, pp. 533–556.

[34] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. 29th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* French Riviera: Springer, 2010, pp. 1–23.

[35] V. Lyubashevsky and D. Micciancio, "Generalized compact knapsacks are collision resistant," in *Proc. 33rd Int. Colloq.* Venice, Italy: Springer, 2016, pp. 144–155.

[36] V. Lyubashevsky, "Digital signatures based on the hardness of ideal lattice problems in all rings," in *Proc. 22nd Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Hanoi, Vietnam: Springer, 2016, pp. 196–214.

[37] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *Proc. 22th Int. Conf. Theory Appl. Cryptograph. Techn.* Warsaw, Poland: Springer, 2003, pp. 614–629.

[38] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *Proc. ACM 6th Conf. LISP Funct. Program.*, Nice, France, 1990, pp. 427–437.

[39] X. Boyen, "Latiice mixing and vanishing trapdoors: A framework for fully secure short signatures and more," in *Proc. 13th Int. Conf. Pract. Theory Public Key Cryptogr.* Paris, France: Springer, 2010, pp. 499–517.

QIN LUO received the B.S. degree in basic mathematics from Jilin University, Changchun, China, in 2014. She is currently pursuing the Ph.D. degree in basic mathematics with Fudan University, Shanghai, China. From 2010 to 2016, her work was the subject of elementary algebra. Later, her major subject is information security and post quantum cryptographic.

CHUN-YANG JIANG studied at the national base class and received the B.S. degree from Jilin University, Changchun, China, in 2014, and the master's degree in statistics from Jilin University. Her work is the research on statistical theory and its applications.