Simpler Efficient Group Signature Scheme with Verifier-Local Revocation from Lattices

Yanhua Zhang[1], Yupu Hu[1], Wen Gao[1] and Mingming Jiang[2]
[1] State Key Laboratory of Integrated Service Networks, Xidian University
Xi'an, 710071 - China
[e-mail: yhzhangxidian@163.com]
[2] School of Computer Science and Technology, Huaibei Normal University
Huaibei, 235000 - China
[e-mail: jiangmm3806586@126.com]
*Corresponding author: Yanhua Zhang

Received May 20, 2015; revised July 27, 2015; revised September 10, 2015; revised October 7, 2015; accepted October 25, 2015; published January 31, 2016

Abstract

Verifier-local revocation (VLR) seems to be the most flexible revocation approaches for any group signature scheme, because it just only requires the verifiers to possess some up-to-date revocation information, but not the signers. Langlois *et al.* (PKC 2014) proposed the first VLR group signature based on lattice assumptions in the random oracle model. Their scheme has at least $\tilde{O}(n^2)\cdot\log N$ bit group public key and $\tilde{O}(n)\cdot\log N$ bit signature, respectively. Here, n is the security parameter and N is the maximum number of group members. In this paper, we present a simpler lattice-based VLR group signature, which is more efficient by a $O(\log N)$ factor in both the group public key and the signature size. The security of our VLR group signature can be reduced to the hardness of learning with errors (LWE) and small integer solution (SIS) in the random oracle model.

Keywords: Group signature, verifier-local revocation, lattice cryptography, LWE, SIS

This paper was supported by the National Natural Science Foundations of China under Grant Nos.61173151 and 61472309.

http://dx.doi.org/10.3837/tiis.2016.01.024 **ISSN : 1976-7277**

1. Introduction

Group signatures have been an active research topic in public-key cryptography since their introduction by Chaum and van Heyst [1]. In a group signature scheme, each group member has a private key that is certified by the manager. By using its private key, each group member can anonymously sign messages on behalf of the whole group (*anonymity*). On the other hand, given a valid group signature σ, the manager should be able to determine which member of the group issued it (*traceability*). These two appealing properties allow group signature schemes to find several real-life applications, such as in trusted computing, digital right management, anonymous online communications, e-commerce systems and much more. Group signatures have proven to be a popular primitive, and since their introduction various constructions based on different assumptions have been proposed [2-8].

The support of membership revocation is a desirable functionality for any group signature. Currently, VLR seems to be the most flexible revocation approaches for any group signature scheme, which only requires the verifiers to possess some up-to-date revocation information, but not the signers. VLR group signatures are implemented by giving the signature verification algorithm an additional argument called the revocation list (*RL*). The *RL* contains a token for each revoked user. The verification algorithm accepts all signatures issued by unrevoked users and reveals no information about which unrevoked user issued the signature. However, if a user is ever revoked (by having its revocation token added to *RL*), signatures from that user are no longer accepted.

In recent years, lattice-based cryptography has attracted significant interest, due to several potential benefits: asymptotic efficiency, security against quantum computers, and worst-case hardness assumptions. Designing secure and efficient lattice-based cryptographic schemes is interesting and challenging. In 2010, Gordon *et al*. [9] made the first step in constructing a secure lattice-based group signature where the sizes of both the group public key and signature were linear in the number of group members N. Later, Laguillaumie *et al*. [10] constructed an efficient lattice-based group signature scheme where the sizes of both the group public key and signature were proportional to $\log N$. However, neither supports the membership revocation. In 2014, Langlois *et al*. [11] constructed a lattice-based group signature with VLR, which was the first lattice-based group signature supporting membership revocation and achieved the same asymptotic efficiency as [10]. Recently, Ling *et al*. [12] and Nguyen *et al*. [13] designed two different efficient lattice-based group signature schemes. By constructing a nice Stern-type non-interactive zero-knowledge proof protocol, the former proposed a scheme which excels previous ones in [10,11] by a constant factor in terms of efficiency, *i.e*., all the sizes are still proportional to $\log N$. Based on a new non-interactive zero-knowledge protocol corresponding to a simple identity-encoding function, the latter also obtained a simpler lattice-based group signature than [9-12], *i.e*., the sizes are shorter by a $O(\log N)$ factor than in the previous works. However, neither of the schemes supports membership revocation. This yields an interesting open problem in this direction: How to construct a simpler and efficient group signature with membership revocation from lattices?

1.1 Our Results

In this paper, we present a new VLR group signature from lattices to reply to the above open problem positively. Compared to [11], it is both simpler and more efficient, saving a $O(\log N)$

factor in both sizes of the group public key and signature. As in [11], our construction satisfies the notion of selfless-anonymity and traceability for the VLR group signatures from [2]. The security of our VLR group signature scheme can be reduced to the hardness of learning with errors (LWE) and small integer solutions (SIS) problem in the random oracle model, which are as hard as several worst-case lattice problems, such as the shortest independent vector problem (SIVP_{γ}) for a polynomial factor $\gamma = poly(n)$.

We give a rough comparison with related lattice-based group signatures in terms of the sizes of the group public-key, the group user secret-key, the signature and whether or not supporting membership revocation in **Table 1**. Here, n denotes the security parameter, and integer N is the maximum number of group members. The other parameters are all implicit function of n.

Table 1. Rough comparison

Schemes	Group public-key	User secret-key	Signature	Membership revocation
[9]	$O(nmN\log q)$	$O(nm\log q)$	$O(nmN\log q)$	No
[10]	$O(nm\log N\log q)$	$O(nm\log q)$	$O(tm\log N\log q)$	No
[11]	$O(nm\log N\log q)$	$O(m\log N\log q)$	$O(tm\log N\log q\log\beta)$	Yes
[12]	$O(nm\log N\log q)$	$O(m\log q)$	$O(tm\log N\log q\log\beta)$	No
[13]	$O(nm\log q)$	$O(nm\log q)$	$O(tm\log q)$	No
Our scheme	$O(nm\log q)$	$O(m\log q)$	$O(tm\log q)$	Yes

1.2 Our Techniques

The first building block of our VLR group signature is an efficient identity-encoding as in [13] to encode the group member's identity by building upon the encoding technique introduced by Agrawal *et al*. [14]. Let the group public key $Gpk=(\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{u})$ consist of three matrices over $\mathrm{Z}_q^{n\times m}$ and a vector in Z_q^n for some positive integers n, m and q. Let $\boldsymbol{A}_i = \left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}\right]$, where i means the group member i. The signing secret key of member i is a short $2m$-dimensional vector $\boldsymbol{x}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}) \in \mathrm{Z}^m \times \mathrm{Z}^m$.

The second building block of our VLR group signature is a non-interactive zero-knowledge proofs of knowledge (NIZKPoK) protocol as in [13] allowing a prover to convince the verifier that it is a certified group member (*i.e.*, it possesses a valid secret signing key). To prove to the verifier that $\boldsymbol{x}_i$ is a short vector in a lattice determined by $\boldsymbol{A}_i$ for some $i \in \{1,2,\cdots,N\}$. Nguyen *et al*. introduced a new problem called split-SIS, which is a variant of SIS and derived a proof of knowledge protocol for a family hash functions as follows:

$$H = \left\{\left(\left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,2}\right], (-\boldsymbol{A}_{2,1}\boldsymbol{x}_{i,2}, \boldsymbol{x}_{i,2}), \beta, N; \boldsymbol{x}_{i,1}, i\right) \in \mathrm{Z}_q^{n\times 2m} \times (\mathrm{Z}_q^n \times \mathrm{Z}^m) \times \mathrm{R} \times \mathrm{Z} \times \mathrm{Z}^m \times \mathrm{Z} : \right.$$
$$\left. \boldsymbol{A}_1\boldsymbol{x}_{i,1} + i\boldsymbol{A}_{2,2}\boldsymbol{x}_{i,2} = -\boldsymbol{A}_{2,1}\boldsymbol{x}_{i,2}, \left\|\boldsymbol{x}_{i,1}\right\| \le \beta\sqrt{m}, i \in \{1,2,\cdots,N\}\right\}$$

The above protocol is repeated many times to make the soundness error negligibly small, then it is transformed into an NIZKPoK using the Fiat-Shamir transformation in the random oracle model.

The third building block of our VLR group signature is a new revocation mechanism. For each group member's secret key $\boldsymbol{x}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})$, let his revocation token be $\boldsymbol{A}_1 \cdot \boldsymbol{x}_{i,1} \bmod q \in \mathrm{Z}_q^n$. By using the Gaussian sampling algorithm and the Bonsai tree principles respectively described in [15,16], we can sample a short vector $(\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}) \in \mathrm{Z}^m \times \mathrm{Z}^m$ from a proper distribution such that

$\boldsymbol{A}_1\boldsymbol{x}_{i,1} + (\boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2})\boldsymbol{x}_{i,2} = \boldsymbol{u}$. And the secret key $\boldsymbol{x}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})$ is statistically indistinguishable for each group number $i \in \{1, 2, \cdots, N\}$. So the revocation token is statistically close to uniform over Z_q^n. For member i, when to sign, he randomly chooses an n-dimensional vector $\boldsymbol{r}_i \leftarrow_R \{-1,1\}^n$, and uses it to construct a cyclic non-invertible square matrix $\boldsymbol{R}_i \in \{-1,1\}^{n\times n}$. Then $\boldsymbol{r}_i$ is appended to the group signature. Our revocation mechanism works as follows: Given a certified group member's signature, the verifier performs the revocation check using a list of tokens of the revoked members $RL = \{grt[k] = \boldsymbol{A}_1 \cdot \boldsymbol{x}_{k,1}\}$. For any two member $i \neq k$, we have that $\boldsymbol{x}_{i,1} \neq \boldsymbol{x}_{k,1}$, and $\boldsymbol{R}_i\boldsymbol{A}_1\boldsymbol{x}_{i,1} \neq \boldsymbol{R}_i\boldsymbol{A}_1\boldsymbol{x}_{k,1}$ with overwhelming probability. Then he checks that if $\boldsymbol{b}_2 = \boldsymbol{R}_i(\boldsymbol{A}_1\boldsymbol{c} - grt[k])$ for all $grt[k] \in RL$. If the equation holds true, which means the member having been revoked, so the verifier rejects the signature. Otherwise, he accepts it.

Putting the above all together, we obtain a simpler efficient VLR group signature scheme from lattices saving a $O(\log N)$ factor in both sizes of the group public key and the signature.

Below, we briefly describe the group signature scheme with VLR from lattices (as depicted in **Fig. 1**).

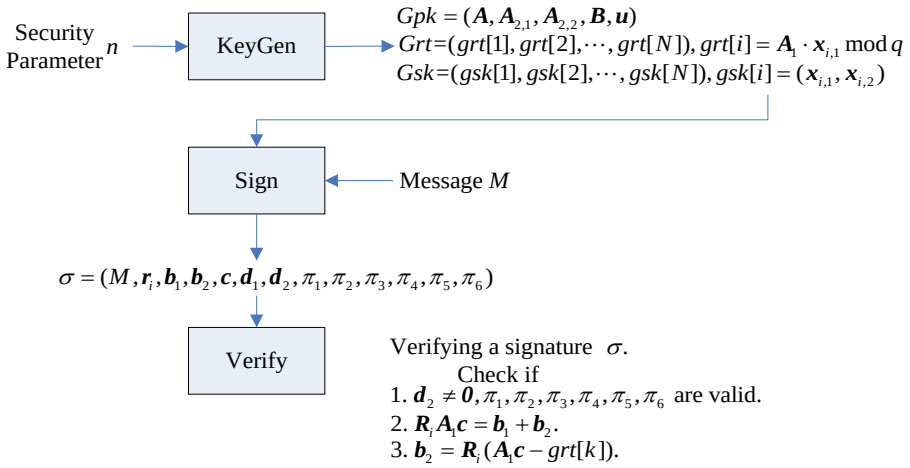


Fig. 1. Simpler efficient group signature scheme with VLR from lattices.

1.3 Outline of this Paper

We introduce some notations, algorithms and several hardness problems on lattices in Section 2. In Section 3, we turn to the VLR group signature, the split-SIS problem and an NIZKPoK protocol for a certified group member. We finally present our construction in Section 4.1, the parameters setting in Section 4.2, the efficiency analysis in Section 4.3, and prove its security of correctness, selfless-anonymous and traceability in Section 4.4.

2. Preliminaries

2.1 Notation

In this paper, the set of real numbers (integers) is denoted by R (Z, *respectively.*). By $\leftarrow_R$, we denote it choosing elements from some distribution uniformly at random. The function log denotes the natural logarithm. Vectors are in column form and denoted by bold lower-case letter (*e.g.*, $\boldsymbol{x}$). The i-th component of $\boldsymbol{x}$ will be denoted by $\boldsymbol{x}_i$. We view a matrix simply as the

set of its column vectors and denoted by bold capital letter (*e.g.*, $\boldsymbol{X}$). The Euclidean norm of $\boldsymbol{x}$ is denoted as $\|\boldsymbol{x}\|$. Define the norm of $\boldsymbol{X}$ as the norm of its longest column (*i.e.*, $\|\boldsymbol{X}\| = \max_i \|\boldsymbol{x}_i\|$). The security parameter throughout this paper is n, and all other quantities are implicit function of n. Let $poly(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant $c > 0$. We use the standard notation of O, ω to classify the growth of functions. If $f(n) = O(g(n) \cdot \log^c n)$, we denote it as $f(n) = \tilde{O}(g(n))$. We use $negl(n)$ to denote a negligible function $f(n) = O(n^{-c})$ for all $c > 0$, and we say a probability is overwhelming if it is 1-$negl(n)$.

2.2 Lattices

Let $\boldsymbol{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m\} \in \mathrm{R}^{m \times m}$ be a matrix with m linearly independent vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m \in \mathrm{R}^m$. The m-dimensional lattice Λ generated by $\boldsymbol{B}$, *i.e.*, $\Lambda = \mathrm{L}(\boldsymbol{B}) = \{\boldsymbol{y} \in \mathrm{R}^m, s.t. \exists \boldsymbol{s} \in \mathrm{Z}^m, \boldsymbol{y} = \sum_{i=1}^m s_i \boldsymbol{b}_i\}$. Here, we focus on integer lattices, *i.e.*, L is contained in Z^m.

Definition 1 For a prime q, a matrix $\boldsymbol{A} \in \mathrm{Z}_q^{n \times m}$ and a vector $\boldsymbol{u}$ in Z_q^n, define:

$$\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) = \{\boldsymbol{e} \in \mathrm{Z}^m \ s.t. \ \boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \bmod q\}, \ \Lambda_q^{\perp}(\boldsymbol{A}) = \{\boldsymbol{e} \in \mathrm{Z}^m \ s.t. \ \boldsymbol{A}\boldsymbol{e} = \boldsymbol{0} \bmod q\}$$

Observe that if $\boldsymbol{t} \in \Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$, then $\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) = \Lambda_q^{\perp}(\boldsymbol{A}) + \boldsymbol{t}$, hence $\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$ is a shift of $\Lambda_q^{\perp}(\boldsymbol{A})$.

Lemma 1 ([17]) Let prime $q \geq 3$ and $m \geq 6n \log q$. There is a probabilistic polynomial-time (PPT) algorithm $\mathrm{TrapGen}(q, n)$ that outputs two matrices $\boldsymbol{A} \in \mathrm{Z}_q^{n \times m}$, $\boldsymbol{T} \in \mathrm{Z}_q^{m \times m}$ such that $\boldsymbol{A}$ is statistically close to uniform matrix in $\mathrm{Z}_q^{n \times m}$ and $\boldsymbol{T}$ is a short basis for $\Lambda_q^{\perp}(\boldsymbol{A})$ satisfying $\|\tilde{\boldsymbol{T}}_{\boldsymbol{A}}\| \leq O(\sqrt{n \log q})$ and $\|\boldsymbol{T}_{\boldsymbol{A}}\| \leq O(n \log q)$ with all but negligible probability in n.

2.3 Discrete Gaussian Distributions

For any $s > 0$, define the Gaussian function on R^m, centered at $\boldsymbol{c}$ with parameter s:

$$\forall \boldsymbol{x} \in \mathrm{R}^m, \rho_{s,\boldsymbol{c}}(\boldsymbol{x}) = exp(-\pi \|\boldsymbol{x} - \boldsymbol{c}\|^2 / s^2)$$

For any $\boldsymbol{c} \in \mathrm{R}^m$, real $s > 0$, m-dimensional lattice Λ, define the Discrete Gaussian Distribution over Λ as:

$$\forall \boldsymbol{x} \in \mathrm{R}^m, D_{\Lambda,s,\boldsymbol{c}}(\boldsymbol{x}) = \frac{\rho_{s,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{s,\boldsymbol{c}}(\Lambda)} = \frac{\rho_{s,\boldsymbol{c}}(\boldsymbol{x})}{\sum_{\boldsymbol{x} \in \Lambda} \rho_{s,\boldsymbol{c}}(\boldsymbol{x})}$$

The subscripts s and $\boldsymbol{c}$ are taken to be 1 and $\boldsymbol{0}$ (*resp.*) when omitted.

Lemma 2 ([15]) Assume that the columns of $\boldsymbol{A} \in \mathrm{Z}_q^{n \times m}$ generate Z_q^n, let $k \in (0, 1/2), s \geq \eta_k(\Lambda^{\perp}(\boldsymbol{A}))$. Then for $\boldsymbol{e} \sim D_{\mathrm{Z}^m,s}$, the distribution of syndrome $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{e} \bmod q$ is within statistical distance $2k$ of uniform over Z_q^n. Furthermore, fix $\boldsymbol{u} \in \mathrm{Z}_q^n$ and let $\boldsymbol{t} \in \mathrm{Z}^m$ be an arbitrary solution to $\boldsymbol{A}\boldsymbol{t} = \boldsymbol{u} \bmod q$. The conditional distribution of $\boldsymbol{e} \sim D_{\mathrm{Z}^m,s}$, given $\boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \bmod q$ is exactly $\boldsymbol{t} + D_{\Lambda^{\perp},s,-\boldsymbol{t}}$.

Definition 2 ([18]) For any m-dimensional lattice Λ and real $k > 0$, the smoothing parameter η_k is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq k$.

Lemma 3 ([18]) Let $q > 2$, $\boldsymbol{A} \in \mathrm{Z}_q^{n \times m}$, $m > n$. Let $\boldsymbol{T} \in \mathrm{Z}_q^{m \times m}$ be a basis for $\Lambda_q^{\perp}(\boldsymbol{A})$, $s \geq \|\tilde{\boldsymbol{T}}\| \cdot \omega(\sqrt{\log m})$. Then for $\boldsymbol{c} \in \mathrm{R}^m, \boldsymbol{u} \in \mathrm{Z}_q^n$:

1. $\Pr_{\boldsymbol{x} \sim D_{\Lambda,s,\boldsymbol{c}}}\left[\|\boldsymbol{x} - \boldsymbol{c}\| > s\sqrt{m}\right] \leq \frac{1+k}{1-k} \cdot 2^{-m}$.

2.There is a PPT algorithm $\mathrm{SampleGau}(\boldsymbol{A},\boldsymbol{T},s,\boldsymbol{c})$ that returns a vector $\boldsymbol{x}\in\Lambda_q^{\perp}(\boldsymbol{A})$ drawn from a distribution statistically close to $D_{\Lambda_q^{\perp}(\boldsymbol{A}),s,\boldsymbol{c}}$.

3.There is a PPT algorithm $\mathrm{SamplePre}(\boldsymbol{A},\boldsymbol{T},\boldsymbol{u},s)$ that returns a vector $\boldsymbol{x}\in\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$ sampled from a distribution statistically close to $D_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}),s}$.

Lemma 4 ([10]) Let $q\geq 3, m>\lceil 6n\log q+n\rceil$, there is a PPT algorithm $\mathrm{SuperSamp}(\boldsymbol{A},\boldsymbol{C})$ taking matrices $\boldsymbol{A}\in \mathrm{Z}_q^{n\times m}$, $\boldsymbol{C}\in \mathrm{Z}_q^{n\times n}$ as inputs, and outputs an almost uniform matrix $\boldsymbol{B}\in \mathrm{Z}_q^{n\times m}$ such that $\boldsymbol{A}\boldsymbol{B}^T=\boldsymbol{C}$, a basis $\boldsymbol{T}_{\boldsymbol{B}}$ of $\Lambda_q^{\perp}(\boldsymbol{B})$ satisfying $\|\boldsymbol{T}_{\boldsymbol{B}}\|\leq m^{1.5}\cdot\omega(\sqrt{\log m})$ and $\|\tilde{\boldsymbol{T}}_{\boldsymbol{B}}\|\leq m\cdot\omega(\sqrt{\log m})$.

Lemma 5 ([16]) Take any matrix $\boldsymbol{A}\in \mathrm{Z}_q^{n\times m_1}$ such that the columns of $\boldsymbol{A}$ span Z_q^n. Let $\boldsymbol{B}\in \mathrm{Z}_q^{n\times m_2}$, and define $\boldsymbol{F}=[\boldsymbol{A}|\boldsymbol{B}]$. There is a polynomial time deterministic algorithm that given $\boldsymbol{A},\boldsymbol{B}$ and a basis $\boldsymbol{T}_{\boldsymbol{A}}$ for $\Lambda_q^{\perp}(\boldsymbol{A})$, and outputs a basis $\boldsymbol{T}_{\boldsymbol{F}}$ for $\Lambda_q^{\perp}(\boldsymbol{F})$ while preserving the Gram-Schmidt norm of the basis (*i.e.*, $\|\tilde{\boldsymbol{T}}_{\boldsymbol{F}}\|=\|\tilde{\boldsymbol{T}}_{\boldsymbol{A}}\|$).

Lemma 6 ([14]) Let $q\geq 3, m>2n\log q$. There is a PPT algorithm $\mathrm{SampleLeft}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{T}_{\boldsymbol{A}},s,\boldsymbol{u})$ taking a full rank matrix $\boldsymbol{A}\in \mathrm{Z}_q^{n\times m}$ a matrix $\boldsymbol{B}\in \mathrm{Z}_q^{n\times m_1}$ a short basis $\boldsymbol{T}_{\boldsymbol{A}}$ for $\Lambda_q^{\perp}(\boldsymbol{A})$, a vector $\boldsymbol{u}\in \mathrm{Z}_q^n$, and a gaussian parameter $s>\|\tilde{\boldsymbol{T}}_{\boldsymbol{A}}\|\cdot\omega(\sqrt{\log(m+m_1)})$ as inputs, and outputs a vector $\boldsymbol{e}\in \mathrm{Z}_q^{m+m_1}$ distributed statistically close to $D_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{F}),s}$, where $\boldsymbol{F}=[\boldsymbol{A}|\boldsymbol{B}]$.

Lemma 7 ([14]) Let $q\geq 3, m>n$. There is a PPT algorithm $\mathrm{SampleRight}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{R},\boldsymbol{T}_{\boldsymbol{B}},s,\boldsymbol{u})$ taking a matrix $\boldsymbol{A}\in \mathrm{Z}_q^{n\times m}$ a full rank matrix $\boldsymbol{B}\in \mathrm{Z}_q^{n\times m}$ a uniform random matrix $\boldsymbol{R}\in\{-1,1\}^{m\times m}$, a short basis $\boldsymbol{T}_{\boldsymbol{B}}$ for $\Lambda_q^{\perp}(\boldsymbol{B})$, a gaussian parameter $s>\|\tilde{\boldsymbol{T}}_{\boldsymbol{B}}\|\cdot\sqrt{m}\cdot\omega(\log m)$ and a vector $\boldsymbol{u}\in \mathrm{Z}_q^n$ as inputs, and outputs a vector $\boldsymbol{e}\in \mathrm{Z}_q^{2m}$ distributed statistically close to $D_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{F}),s}$, where $\boldsymbol{F}=[\boldsymbol{A}|\boldsymbol{A}\boldsymbol{R}+\boldsymbol{B}]$.

Lemma 8 ([14]) Let $\boldsymbol{R}$ be an $m\times m$ matrix chosen at random from $\{-1,1\}^{m\times m}$, then for all vectors $\boldsymbol{u}\in \mathrm{R}^m$, we have that $\Pr\left[\|\boldsymbol{R}\boldsymbol{u}\|>\|\boldsymbol{u}\|\sqrt{m}\cdot\omega(\sqrt{\log m})\right]<negl(m)$.

2.4 Hardness Assumption

The learning with errors (LWE) problem defined by Regev [19] was suggested to be a classic hard problem on lattice.

Definition 3 ([19]) Let $n\in \mathrm{Z}$ and $q=q(n)$ be positive integers, $\alpha\in \mathrm{R}$ be a positive real, χ_α be some Discrete Gaussian Distribution over Z_q^m. Define $\boldsymbol{A}_{\boldsymbol{s},\chi_\alpha}\subseteq \mathrm{Z}_q^{n\times m}\times \mathrm{Z}_q^m$ as the distribution of $(\boldsymbol{A},\boldsymbol{A}^T\boldsymbol{s}+\boldsymbol{x})$, where $\boldsymbol{A}\leftarrow_R \mathrm{Z}_q^{n\times m}, \boldsymbol{s}\leftarrow_R \mathrm{Z}_q^n, \boldsymbol{x}\leftarrow_R \chi_\alpha$. An algorithm solves $\mathrm{LWE}_{q,\chi_\alpha}$ if for randomly chosen $\boldsymbol{s}\in \mathrm{Z}_q^n$, given a sample from $\boldsymbol{A}_{\boldsymbol{s},\chi_\alpha}$, it outputs $\boldsymbol{s}\in \mathrm{Z}_q^n$ with overwhelming probability. The decisional variant of $\mathrm{LWE}_{q,\chi_\alpha}$ is that, for a uniformly chosen $\boldsymbol{s}\in \mathrm{Z}_q^n$, an algorithm is asked to distinguish $\boldsymbol{A}_{\boldsymbol{s},\chi_\alpha}$ from the uniform distribution over $\mathrm{Z}_q^{n\times m}\times \mathrm{Z}_q^m$.

Lemma 9 ([19]) Let $\alpha=\alpha(n)\in(0,1)$ and let $q=q(n)$ be a prime such that $\alpha q>2\sqrt{n}$. If there is an efficient (possibly quantum) algorithm that solves $\mathrm{LWE}_{q,\chi_\alpha}$, then there is an efficient quantum algorithm for approximating shortest independent vector problem (SIVP) in the Euclidean norm, and in the worst-case to within $\tilde{O}(n/\alpha)$ factors.

The small integer solution (SIS) problem was suggested to be hard and formally defined by

Micciancio and Regev [18].

Definition 4 ([18]) The SIS problem in Euclidean norm is: given a prime q, a matrix $\boldsymbol{A} \in \mathrm{Z}_q^{n\times m}$ and a real β, find a non-zero vector $\boldsymbol{e} \in \mathrm{Z}^m$ such that $\boldsymbol{Ae} = \boldsymbol{0} \bmod q$ and $\|\boldsymbol{e}\| \le \beta$. The average-case $\mathrm{SIS}_{q,m,\beta}$ problem is defined similarly, where $\boldsymbol{A} \in \mathrm{Z}_q^{n\times m}$ is uniformly random.

Lemma 10 ([15]) For poly-bounded m, any $\beta = poly(n)$ and for any prime $q \ge \beta\cdot\omega(\sqrt{n\log n})$, the average-case $\mathrm{SIS}_{q,m,\beta}$ problem is as hard as approximating SIVP problem, among others, in the worst-case to within $\gamma = \beta\cdot\tilde{O}(\sqrt{n})$ factors.

2.5 Non-interactive Zero-Knowledge Proofs of Knowledge

In 2013, Laguillaumie *et al*. [10] gave a non-interactive zero-knowledge proof of knowledge (NIZKPoK) for the inhomogeneous small integer solution (ISIS) relations:

$$R_{\mathrm{ISIS}} = \left\{(\boldsymbol{A}, \boldsymbol{y}, \beta; \boldsymbol{x}) \in \mathrm{Z}_q^{n\times m} \times \mathrm{Z}_q^n \times \mathrm{R} \times \mathrm{Z}^m s.t.\ \boldsymbol{Ax} = \boldsymbol{y} \text{ and } \|\boldsymbol{x}\| \le \beta\right\}$$

By using the duality between LWE and ISIS, there is an NIZKPoK for the LWE relations:

$$R_{\mathrm{LWE}} = \left\{(\boldsymbol{A}, \boldsymbol{b}, \alpha; \boldsymbol{s}) \in \mathrm{Z}_q^{n\times m} \times \mathrm{Z}_q^m \times \mathrm{R} \times \mathrm{Z}_q^n s.t.\ \|\boldsymbol{b} - \boldsymbol{A}^T\boldsymbol{s}\| \le \alpha q\sqrt{m}\right\}$$

In 2015, Nguyen *et al*. [13] gave an NIZKPoK for the extended-LWE (eLWE) relations:

$$R_{\mathrm{eLWE}} = \left\{(\boldsymbol{A}, \boldsymbol{b}, \gamma; \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{x}) \in \mathrm{Z}_q^{n\times m} \times \mathrm{Z}_q^m \times \mathrm{R} \times \mathrm{Z}_q^n \times \mathrm{Z}^{2m} s.t.\ \boldsymbol{b} = \boldsymbol{A}^T\boldsymbol{s} + \boldsymbol{e} + \boldsymbol{x}, \text{and } \|\boldsymbol{e}\| \le \gamma \text{ and } \|\boldsymbol{x}\| \le \gamma\right\}$$

3. VLR Group Signature

In this section, we formalize the definition and security model of VLR group signature [2]. Then we turn to review the split-SIS problem and an NIZKPoK protocol for a certified group member as in [13].

3.1 VLR Group Signature

A verifier-local revocation (VLR) group signature scheme consists of three algorithms, that is, KeyGen, Sign and Verify, and all of them are described as follows:

KeyGen(n, N): A PPT algorithm takes the security parameter n and the number of group members N as inputs, outputs a group public key Gpk, an N-dimensional vector of members secret key $Gsk=(gsk[1], gsk[2], \cdots, gsk[N])$, and an N-dimensional vector of members revocation tokens $Grt=(grt[1], grt[2], \cdots, grt[N])$.

Sign(Gpk, $gsk[i]$, M): A PPT algorithm takes the group public key Gpk, a member signing secret key $gsk[i]$, and a message $M \in \{0,1\}^*$ as inputs, outputs a signature σ.

Verify(Gpk, RL, σ, M): A deterministic algorithm takes the group public key Gpk, a set of revocation tokens $RL \subseteq Grt$, a signature σ, and the message M as inputs, outputs either "Valid" or "Invalid". The output "Valid" indicates that σ is a valid signature on message M under Gpk, and the signer has not been revoked.

Any VLR group signature has an implicit tracing algorithm that, using Grt as the tracing key. Given a valid message-signature pair (M, σ), a party possessing Grt can determine the signer of σ by running $\mathrm{Verify}(Gpk, RL = grt[i], \sigma, M)$, for $i = 1, 2, \cdots$, and outputting the first index $i^* \in \{1, 2, \cdots, N\}$ for which the verification algorithm return "Invalid".

A secure VLR group signature scheme must satisfy the following requirements: correctness, selfness-anonymity and traceability. All are described as follows:

Correctness: For(*Gpk*, *Gsk*, *Grt*) generated by KeyGen, every signature generated by a group member $i \in \{1,2,\cdots,N\}$: Verify(*Gpk*, *RL*, Sign(*Gpk*, *gsk*[*i*], *M*) = Valid $\Leftrightarrow$ *gsk*[*i*] $\notin$ *RL*.

Selfless-anonymity: In this game, the adversary A's goal is is to determine which of the two adaptively chosen keys generated a signature. A is not given access to either key. The game is formulated as follows:

(a) **Setup**: The challenger runs KeyGen, obtaining (*Gpk*, *Gsk*, *Grt*), then provides *Gpk* to A.

(b) **Queries**: A can make the queries as follows:

Signing: Request for a signature on the message $M \in \{0,1\}^*$ of member *i*. The challenger returns σ, where $\sigma = \text{Sign}(Gpk, gsk[i], M)$.

Corruption: Request for the secret key of member *i*. The challenger returns *gsk*[*i*].

Revocation: Request for the revocation token of member *i*.The challenger returns *grt*[*i*].

(c) **Challenge**: A outputs a message *M* and two members i_0 and i_1. It must have made neither a corruption or revocation query at either index. The challenger chooses a bit $b \leftarrow_R \{0,1\}$, and computes a signature on *M* by i_b as $\sigma^* = \text{Sign}(Gpk, gsk[i_b], M)$, then returns σ^* to A.

(d) **Restricted Queries**: After obtaining the challenge, A can still make queries as before, but with restrictions as follows: it is not allowed to make any corruption or revocation query for member i_0 or i_1.

(e) **Output**: Finally, A outputs a bit *b*'. It wins the game if *b*' = *b*.

We define the adversary's advantage in winning the game as $\text{Adv}_A = |\Pr(b' = b) - 1/2|$. We say that the VLR group signature is selfless-anonymous if Adv_A is negligible.

Traceability: In this game, A's goal is to forge a signature that cannot be traced to one of the members in its coalition using the tracing algorithm. The game is formulated as follows:

(a) **Setup**: The challenger runs KeyGen, obtaining (*Gpk*, *Gsk*, *Grt*), then provides *Gpk* and *Grt* to A. Set the corruption set $U = \varnothing$.

(b) **Queries**: A can make the queries as follows:

Signing: Request for a signature on $M \in \{0,1\}^*$ of member *i*. The challenger returns σ, where $\sigma = \text{Sign}(Gpk, gsk[i], M)$.

Corruption: Request for the secret key of member *i*.The challenger appends *i* to the set *U*, and returns *gsk*[*i*].

(c) **Forgery**: Finally, A outputs a message M^*, a set of revocation tokens RL^* and a signature σ^*. The adversary wins the game if:

1. Verify$(Gpk, RL^*, \sigma^*, M^*)$ = Valid.
2. The (implicit) tracing algorithm fails or traces to a member outside of the coalition $U \setminus RL^*$.
3. The signature σ^* is non-trivial, *i.e.*, A did not obtain σ^* by making a signing query on M^*.

We denote by SuccPT_A the probability that adversary A wins the above game. And we say that the VLR group signature is traceable if SuccPT_A is negligible.

3.2 Split-SIS Problems

The Split-SIS problem was suggested to be hard on lattices defined by Nguyen *et al*. [13].

Definition 5 ([13]) Given uniformly random matrices $(\boldsymbol{A}_1, \boldsymbol{A}_2) \in Z_q^{n\times m} \times Z_q^{n\times m}$, integer *N* and β, an algorithm solving the $\text{Split-SIS}_{q,m,\beta,N}$ problem is asked to output a tuple $(\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2), h) \in Z^{2m} \times Z$ such that: i. $\boldsymbol{x}_1 \neq \boldsymbol{0}$ or $h\boldsymbol{x}_2 \neq \boldsymbol{0}$. ii. $\|\boldsymbol{x}\| \leq \beta, h \in \{1,2,\cdots,N\}$, and $\boldsymbol{A}_1\boldsymbol{x}_1 + h\boldsymbol{A}_2\boldsymbol{x}_2 = \boldsymbol{0} \bmod q$.

Lemma 11 ([13]) For poly-bounded m, $\beta = poly(n)$, $N = N(n)$, and prime $q \geq \beta \cdot \omega(\sqrt{n \log n}) > N$, the $\text{Split-SIS}_{q,m,\beta,N}$ problem is equivalent to the $\text{SIS}_{q,2m,\beta}$ problem. In particular, the average-case $\text{Split-SIS}_{q,m,\beta,N}$ problem is as hard as approximating SIVP problem, in the worst-case to within $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ factors.

3.3 An NIZKPoK Protocol

In this subsection, we present an NIZKPoK protocol as in [13] allowing a prover to convince the verifier that it is a certified group member.

For any $i \in \{1, 2, \cdots, N\}$, we compute the representation of i in base $\overline{\beta} = \lfloor \beta \rfloor$, a l-dimensional vector $\boldsymbol{v}_i = (v_1, v_2, \cdots, v_l) \in \mathrm{Z}^l$ such that $0 \leq v_b \leq \beta - 1$, and $i = \sum_{b=1}^{l} v_b \overline{\beta}^b$, where $l = \lceil \log_{\overline{\beta}} N \rceil$. And we denote $\boldsymbol{b} = \boldsymbol{A}_2 \boldsymbol{x}_{i,2}$, and compute $\boldsymbol{D} = (\boldsymbol{b}, \overline{\beta}\boldsymbol{b}, \cdots, \overline{\beta}^{l-1}\boldsymbol{b}) \in \mathrm{Z}_q^{n \times l}$.

We first present a interactive zero-knowledge protocol to prove that $\boldsymbol{x} = (\boldsymbol{x}_{i,1}, \boldsymbol{v}_i) \in \mathrm{Z}^m \times \mathrm{Z}^l$ is a short vector such that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \bmod q$, where $\boldsymbol{A} = \left[\boldsymbol{A}_1 \middle| \boldsymbol{D} \right] \in \mathrm{Z}_q^{n \times (m+l)}$. The protocol makes use of the rejection sampling technique to achieve zero-knowledge with single-bit challenge $t = \omega(\log n)$ times in parallel.

Let $\gamma = \eta \cdot m^{1.5}$, $\zeta(\boldsymbol{z}, \boldsymbol{y}) = 1 - \min(\frac{D_{\mathrm{Z}^{m+l},\gamma}(\boldsymbol{z})}{M_l \cdot D_{\mathrm{Z}^{m+l},\boldsymbol{y},\gamma}(\boldsymbol{z})}, 1)$, where $\boldsymbol{y}, \boldsymbol{z} \in \mathrm{Z}^{m+l}$, and $M_l \leq 1 + O(\frac{1}{m})$ is set according to [20], then the protocol is described as follows:

1.The prover P generates a commitment $\text{Com}=(\boldsymbol{u}_k)_{k \in \{1,2,\cdots,t\}}$, where for each k, $\boldsymbol{u}_k \in \mathrm{Z}_q^n$ is obtained by sampling $\boldsymbol{e}_k \leftarrow_R D_{\mathrm{Z}^{m+l},\gamma}$ and computing $\boldsymbol{u}_k = \boldsymbol{A}\boldsymbol{e}_k \bmod q$. Then, Com is sent to the verifier V.

2. The verifier V sends a challenge $\text{Chall}=(c_k)_{k \in \{1,2,\cdots,t\}} \leftarrow_R \{0,1\}$ to P.

3. For $k \in \{1, 2, \cdots, t\}$, the prover P does as follows:

i. Compute $\boldsymbol{z}_k = \boldsymbol{e}_k + c_k \cdot \boldsymbol{x}$.

ii. Set $\boldsymbol{z}_k = \perp$ with probability $\zeta(\boldsymbol{z}, \boldsymbol{y})$. Then $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_t)$ is sent to V.

4. For $k \in \{1, 2, \cdots, t\}$, the verifier V checks as follows:

i. Set $d_k = 1$, if $\|\boldsymbol{z}_k\| \leq 2\gamma\sqrt{m+l}$ and $\boldsymbol{A}\boldsymbol{z}_k = \boldsymbol{u}_k + c_k \cdot \boldsymbol{y} \bmod q$. Otherwise, set $d_k = 0$.

ii. Return 1 and accept if and only if $\sum_{k \in \{1,2,\cdots,t\}} d_k \geq 0.65t$.

Since the binary challenges are used, the protocol has the property of special soundness. Given two transcripts (Com, Chall, $\boldsymbol{z}$) and (Com, Chall′, $\boldsymbol{z}'$) with distinct challenge $\text{Chall} \neq \text{Chall}'$, one can extract a "weak" witness $\boldsymbol{x}' = \boldsymbol{z}_k - \boldsymbol{z}_k'$ for some k satisfying $\boldsymbol{A}\boldsymbol{x}' = \boldsymbol{y}$ and $\|\boldsymbol{x}'\| \leq 4\gamma\sqrt{2m}$.

Applying the Fiat-Shamir transformation in a standard way, one can obtain an NIZKPoK by computing $\text{Chall}=H(\rho, \text{Com})$, where $H: \{0,1\}^* \rightarrow \{0,1\}^t$ is modeled as a random oracle, and ρ represents all the other auxiliary inputs, *e.g.*, a specified message M to be signed.

4. A Simple and Efficient VLR Group Signature from Lattices

4.1 Our Construction

Inspired by the work of [13], we now construct a simple and efficient lattice-based VLR group

signature. The main steps of our VLR group signature scheme are provided as follows:

KeyGen(n, N): Take a security parameter n and the number of group members N as inputs, set the parameters $m, q, s, \alpha, \beta, \eta$ as specified in Section 4.2 below, and choose a hash function $H: \{0,1\}^* \to \{0,1\}^t$ for the NIZKPoK proof, where $t = \omega(\log n)$. Then, do these steps as follows:

1. Using $\mathrm{TrapGen}(q, n)$, the group manager generates a matrix $\boldsymbol{A}_1 \in \mathrm{Z}_q^{n \times m}$ together with a short basis $\boldsymbol{T}_{\boldsymbol{A}_1} \in \mathrm{Z}_q^{m \times m}$ for $\Lambda_q^{\perp}(\boldsymbol{A}_1)$, and randomly chooses $\boldsymbol{u} \in \mathrm{Z}_q^n$, $\boldsymbol{A}_{2,1}$, $\boldsymbol{A}_{2,2} \in \mathrm{Z}_q^{n \times m}$.
2. Using $\mathrm{SuperSamp}(\boldsymbol{A}_1, \boldsymbol{0})$, the group manager generates a matrix $\boldsymbol{B} \in \mathrm{Z}_q^{n \times m}$ satisfying $\boldsymbol{A}_1 \boldsymbol{B}^T = \boldsymbol{0}$.
3. For $i \in \{1, \cdots, N\}$, define $\boldsymbol{A}_i = \left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2} \right]$. The group member i chooses $\boldsymbol{x}_{i,2} \leftarrow_R D_{\mathrm{Z}^m, \beta}$, and computes $(\boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}) \cdot \boldsymbol{x}_{i,2} = \boldsymbol{u}_i$. Using $\mathrm{SampleLeft}(\boldsymbol{A}_1, \boldsymbol{T}_{\boldsymbol{A}_1}, \beta, \boldsymbol{u} - \boldsymbol{u}_i)$, the manager generates a short vector $\boldsymbol{x}_{i,1} \in \mathrm{Z}_q^m$, such that $\boldsymbol{A}_1 \cdot \boldsymbol{x}_{i,1} + (\boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}) \cdot \boldsymbol{x}_{i,2} = \boldsymbol{u}$.
4. Let the group member i's secret key be $gsk[i] = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}) \in \mathrm{Z}^m \times \mathrm{Z}^m$, and its revocation token be $grt[i] = \boldsymbol{A}_1 \cdot \boldsymbol{x}_{i,1} \bmod q \in \mathrm{Z}_q^n$.
5. Finally, output the group public key $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u})$, the group members secret key $Gsk = (gsk[1], gsk[2], \cdots, gsk[N])$, and the revocation tokens $Grt = (grt[1], grt[2], \cdots, grt[N])$.

Sign(Gpk, $gsk[i]$, M): Take the group public key $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u})$, the member i's secret key $gsk[i] = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})$, and a message $M \in \{0,1\}^*$ as inputs, the member i does these steps as follows:

1. Randomly choose $\boldsymbol{s} \leftarrow_R \mathrm{Z}_q^n$, $\boldsymbol{e} \leftarrow_R \chi_\alpha$, and an n-dimensional vector $\boldsymbol{r}_i \leftarrow_R \{-1,1\}^n$. Using $\boldsymbol{r}_i$ to construct a cyclic square matrix $\boldsymbol{R}_i \in \{-1,1\}^{n \times n}$. If $\boldsymbol{R}_i$ is invertible, then simply re-choose $\boldsymbol{r}_i$. Let $\boldsymbol{b}_1 = \boldsymbol{R}_i \boldsymbol{A}_1 \boldsymbol{x}_{i,1} \in \mathrm{Z}_q^n, \boldsymbol{b}_2 = \boldsymbol{R}_i \boldsymbol{A}_1 \boldsymbol{e} \in \mathrm{Z}_q^n$, and $\boldsymbol{c} = \boldsymbol{B}^{\mathrm{T}} \boldsymbol{s} + \boldsymbol{e} + \boldsymbol{x}_{i,1} \in \mathrm{Z}^m$.
2. Generate an NIZKPoK π_1 of $(\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{x}_{i,1})$ such that $(\boldsymbol{B}, \boldsymbol{c}, \eta; \boldsymbol{s}, \boldsymbol{e}, \boldsymbol{x}_{i,1}) \in R_{\mathrm{eLWE}}$, and two NIZKPoKs π_2, π_3 of $\boldsymbol{x}_{i,1}$ such that $(\boldsymbol{R}_i \boldsymbol{A}_1, \boldsymbol{b}_1, \beta; \boldsymbol{x}_{i,1}) \in R_{\mathrm{ISIS}}, (\boldsymbol{R}_i \boldsymbol{A}_1, \boldsymbol{b}_2, \beta; \boldsymbol{e}) \in R_{\mathrm{ISIS}}$.
3. Let $\boldsymbol{d}_1 = \boldsymbol{A}_{2,1} \boldsymbol{x}_{i,2} \in \mathrm{Z}_q^n, \boldsymbol{d}_2 = \boldsymbol{A}_{2,2} \boldsymbol{x}_{i,2} \in \mathrm{Z}_q^n$. Then generate two NIZKPoKs π_4, π_5 of $\boldsymbol{x}_{i,2}$ such that $(\boldsymbol{A}_{2,1}, \boldsymbol{d}_1, \beta; \boldsymbol{x}_{i,2}) \in R_{\mathrm{ISIS}}, (\boldsymbol{A}_{2,1}, \boldsymbol{d}_2, \beta; \boldsymbol{x}_{i,2}) \in R_{\mathrm{ISIS}}$.
4. Let $\overline{\beta} = \lfloor \beta \rfloor, l = \left\lceil \log_{\overline{\beta}} N \right\rceil$, and $\boldsymbol{D} = (\boldsymbol{d}_2, \overline{\beta} \boldsymbol{d}_2, \cdots, \overline{\beta}^{l-1} \boldsymbol{d}_2) \in \mathrm{Z}_q^{n \times l}$. Then, generate an NIZKPoK π_6 of $\boldsymbol{x}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})$, $\boldsymbol{e}$ and $\boldsymbol{v}_i = (v_1, v_2, \cdots, v_l) \in \mathrm{Z}_{\overline{\beta}}^l$ of $i \in \{1, 2, \cdots, N\}$ such that,

$$\boldsymbol{A}_1 \boldsymbol{c} + \boldsymbol{d}_1 = \boldsymbol{A}_1 \boldsymbol{e} + \boldsymbol{u} - \boldsymbol{D} \boldsymbol{v}_i, \boldsymbol{A}_1 \boldsymbol{c} = \boldsymbol{A}_1 \boldsymbol{e} + \boldsymbol{A}_1 \boldsymbol{x}_{i,1}$$

Where the challenge is computed by $H(\boldsymbol{c}, \boldsymbol{b}_1, \boldsymbol{b}_2, \pi_1, \pi_4, \pi_5, M, \mathrm{Com})$, and Com is the commitment message for π_6. (The proof π_6 is given in the appendix A.)
5. Output the VLR group signature $\sigma = (M, \boldsymbol{r}_i, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{c}, \boldsymbol{d}_1, \boldsymbol{d}_2, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6)$.

Verify(Gpk, RL, σ, M): Take the group public key $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u})$, a set of tokens $RL = (grt[k])_{k \le N} \subseteq Grt$ whose cardinality is at most N-1. Do these steps as follows:

1. Parse the signature $\sigma = (M, \boldsymbol{r}_i, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{c}, \boldsymbol{d}_1, \boldsymbol{d}_2, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6)$.
2. Check if $\boldsymbol{d}_2 \neq \boldsymbol{0}$, and the proofs $\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6$ are valid.
3. Use $\boldsymbol{r}_i$ to construct $\boldsymbol{R}_i$, and check if $\boldsymbol{R}_i \boldsymbol{A}_1 \boldsymbol{c} = \boldsymbol{b}_1 + \boldsymbol{b}_2$.
4. If the above all are satisfied, then accept σ is outputted by a certified group member.
5. Check if $\boldsymbol{b}_2 = \boldsymbol{R}_i (\boldsymbol{A}_1 \boldsymbol{c} - grt[k])$. If the equation holds true, then output "Invalid". Otherwise,

output "Valid" and accept σ outputted by a certified group member who has not been revoked.

4.2 Parameters

Our scheme depends on several parameters $m, q, s, \alpha, \beta, \eta$, which are set so that all algorithms can be implemented in polynomial time and correct, and the security properties hold. Assume that the security parameter is n, all other parameters are determined as follows:

$$m = 6\lceil (n+1)\log q + n \rceil, s = m \cdot \omega(\log m), \beta = m^{1.5} \cdot \omega(\log^{1.5} m),$$
$$q = m^{2.5} \max(m^6 \cdot \omega(\log^{2.5} m), 4N), \alpha = 2\sqrt{m}/q, \eta = m^2 \cdot \omega(\log^{1.5} m).$$

4.3 Efficiency

The parameters are set in Section 4.2 so that all algorithms in our VLR group signature can be implemented in polynomial time. Since the group public key only contains four matrices over $Z_q^{n\times m}$, and a vector in Z_q^n, it has bit-size $4nm\log q + n\log q = O(nm\log q)$. For the bit-length of the signature $\sigma = (M, \boldsymbol{r}_i, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{c}, \boldsymbol{d}_1, \boldsymbol{d}_2, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6)$, we know that the bit-length of $\boldsymbol{r}_i$ and $\boldsymbol{c}$ are at most $2n$ and $m\log q$, respectively. All of the bit-length of $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{d}_1, \boldsymbol{d}_2$ are at most $n\log q$. If we set the repetition parameter $t = \omega(\log n)$ for the proofs $\pi_1, \pi_2, \pi_3, \pi_4, \pi_5$ and π_6. The bit-length of π_1 is at most $(3m-n)t\log q$, and the bit-length $\pi_2, \pi_3, \pi_4, \pi_5$ are at most $(m+n)t\log q$, respectively. The bit-length of π_6 is at most $(2m+2n+l)t\log q$. Since $l = \lceil \log_{\bar{\beta}} N \rceil < n$ and $m = O(n\log q)$, thus, the total bit-size of the signature σ is as follows:

$$2n + m\log q + 4n\log q + (3m-n)t\log q + 4(m+n)t\log q + (2m+2n+l)t\log q$$
$$= 2n + (m+4n)\log q + (9m+5n+l)t\log q = O(tm\log q).$$

4.4 The Security

In this subsection, we focus on the correctness, selfless-anonymous and traceability of our VLR group signature scheme.

Theorem 1 (Correctness) Our scheme is correct with overwhelming probability.

Proof For $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u}), Gsk = (gsk[1], \cdots, gsk[N]), Grt = (grt[1], \cdots, grt[N])$ outputted by KeyGen(n, N), all $i \in \{1, 2, \cdots, N\}$, and $M \in \{0,1\}^*$, we have to prove that:

$$\text{Verify}(Gpk, RL, \text{Sign}(Gpk, gsk[i], M) = \text{Valid} \Leftrightarrow gsk[i] \notin RL.$$

1. We first prove that: $gsk[i] \notin RL \Rightarrow \text{Verify}(Gpk, RL, \text{Sign}(Gpk, gsk[i], M) = \text{Valid}$.

Suppose that $gsk[i] \notin RL$, the verifier computes $\boldsymbol{R}_i(\boldsymbol{A}_1\boldsymbol{c} - grt[k]) = \boldsymbol{R}_i(\boldsymbol{A}_1\boldsymbol{e} + \boldsymbol{A}_1\boldsymbol{x}_{i,1} - \boldsymbol{A}_1\boldsymbol{x}_{k,1})$. And because $gsk[i] \notin RL, \boldsymbol{A}_1\boldsymbol{x}_{i,1} - \boldsymbol{A}_1\boldsymbol{x}_{k,1} \neq \boldsymbol{0}$, so $\boldsymbol{R}_i(\boldsymbol{A}_1\boldsymbol{c} - grt[k]) \neq \boldsymbol{R}_i\boldsymbol{A}_1\boldsymbol{e}$ with overwhelming probability. Hence, $\text{Verify}(Gpk, RL, \text{Sign}(Gpk, gsk[i], M) = \text{Valid}$.

2. We then prove that: $\text{Verify}(Gpk, RL, \text{Sign}(Gpk, gsk[i], M) = \text{Valid} \Rightarrow gsk[i] \notin RL$.

Suppose that $gsk[i] = \boldsymbol{A}_1\boldsymbol{x}_{i,1} \bmod q \in RL$, so $\boldsymbol{R}_i\boldsymbol{A}_1\boldsymbol{e} = \boldsymbol{R}_i(\boldsymbol{A}_1\boldsymbol{c} - grt[k])$ for $k=i$. On the other hand, since $\text{Verify}(Gpk, RL, \text{Sign}(Gpk, gsk[i], M) = \text{Valid}$, we have that $\boldsymbol{R}_i\boldsymbol{A}_1\boldsymbol{e} \neq \boldsymbol{R}_i(\boldsymbol{A}_1\boldsymbol{c} - grt[k])$. Thus, we obtain a contradiction. Namely, $gsk[i] \notin RL$. This concludes the correctness proof.

Theorem 2 (Selfless-anonymity) Under the LWE assumption, our VLR group signature is selfless-anonymity in the random oracle model.

Proof We define a sequence of hybrid games as follows:

Game G_0: G_0 is the original selfless-anonymity game. The challenger honestly does as follows:

Run KeyGen(n, N) to obtain that $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u})$, Gsk=(gsk[1], gsk[2],$\cdots$, gsk[N]), and Grt=(grt[1], grt[2],$\cdots$, grt[N]). Then set $RL = \varnothing$, $Corrupted$=$\varnothing$, and give Gpk to adversary A.

2. If A queries the signature on any message M of member i, return $\sigma = \mathrm{Sign}(Gpk, gsk[i], M)$.

If A queries the corruption of member i, set $Corrupted$=$Corrupted$ $\cup\{i\}$, and return gsk[i].

If A queries the revocation of member i, set $RL = RL \cup \{grt[i]\}$, and return grt[i].

3. A outputs a message M^* and two members i_0 and i_1 such that $i_b \notin Corrupted$ and $grt[i_b] \notin RL$ for each $b \in \{0,1\}$.

4. Pick a bit $b \leftarrow_R \{0,1\}$, and generate a valid VLR group signature

$$\sigma^* = \mathrm{Sign}(Gpk, gsk[i_b], M^*) = (M^*, \boldsymbol{r}_{i_b}^*, \boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \boldsymbol{c}^*, \boldsymbol{d}_1^*, \boldsymbol{d}_2^*, \pi_1^*, \pi_2^*, \pi_3^*, \pi_4^*, \pi_5^*, \pi_6^*)$$

and return σ^* to A.

5. A can still make queries as before, but not allowed to ask for $gsk[i_b]$ or $grt[i_b]$ for $b \in \{0,1\}$.

6. Finally, A outputs a bit b'.

Game G_1: In this game, the challenger does the same as in G_0, except that it uses the NIZKPoK simulators(by appropriately programming the random oracle) to generate $(\pi_1^*, \pi_2^*, \pi_3^*, \pi_4^*, \pi_5^*, \pi_6^*)$. By the property of NIZKPoKs, G_1 is computationally indistinguishable from G_0.

Game G_2: In G_2, the challenger does the same as in G_1, except that it computes $\boldsymbol{c} = \boldsymbol{u} + \boldsymbol{x}_{i_b,1}^*$ with a randomly chosen $\boldsymbol{u} \leftarrow_R \mathrm{Z}_q^m$.

Lemma 12 Under the LWE assumption, G_2 is computationally indistinguishable from G_1.

Proof Assume there is an algorithm A which distinguishes G_2 from G_1 with non-negligible probability. Using A, there is an algorithm C that breaks the LWE assumption. Given a LWE tuple $(\hat{\boldsymbol{B}}, \hat{\boldsymbol{u}}) \in \mathrm{Z}_q^{n\times m} \times \mathrm{Z}_q^m$, C sets $\boldsymbol{B} = \hat{\boldsymbol{B}}$ and uses $\mathrm{SuperSamp}(\boldsymbol{B}, \boldsymbol{0})$ to obtain a matrix $\boldsymbol{A}_1$ together with a short basis $\boldsymbol{T}_{\boldsymbol{A}_1}$ for $\Lambda_q^{\perp}(\boldsymbol{A}_1)$. Then, it chooses $\boldsymbol{u} \in \mathrm{Z}_q^n$, $\boldsymbol{A}_{2.1}, \boldsymbol{A}_{2.2} \in \mathrm{Z}_q^{n\times m}$. For $i \in \{1,\cdots,N\}$, and define $\boldsymbol{A}_i = \left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}\right]$. The member i chooses $\boldsymbol{x}_{i,2} \leftarrow_R D_{\mathrm{Z}^m,\beta}$, and computes $(\boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}) \cdot \boldsymbol{x}_{i,2} = \boldsymbol{u}_i$. C uses $\mathrm{SampleLeft}(\boldsymbol{A}_1, \boldsymbol{T}_{\boldsymbol{A}_1}, \beta, \boldsymbol{u} - \boldsymbol{u}_i)$ to obtain a short vector $\boldsymbol{x}_{i,1} \in \mathrm{Z}_q^m$. The member i's secret key be $gsk[i] = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2})$ and revocation token be $grt[i] = \boldsymbol{A}_1 \cdot \boldsymbol{x}_{i,1} \bmod q \in \mathrm{Z}_q^n$. Finally, C outputs the group public key $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u})$, the members secret key Gsk=(gsk[1], gsk[2],$\cdots$, gsk[N]), and the revocation token Grt=(grt[1], grt[2],$\cdots$, grt[N]) to A. So the distributions of Gpk, Gsk, Grt are statistically close to that in G_0 and G_1.

When generating the challenge signature, C does the same as in G_1, except that it computes $\boldsymbol{c}^* = \hat{\boldsymbol{u}} + \boldsymbol{x}_{i_b,1}^*$. If $(\hat{\boldsymbol{B}}, \hat{\boldsymbol{u}})$ is a LWE tuple with respect to χ_α, $\boldsymbol{c}$ is the same as in G_1. Otherwise, we have that $\hat{\boldsymbol{u}}$ is uniformly distributed over Z_q^m, which is as in G_2. If A can distinguish G_2 from G_1 with advantage ò, then C can break the LWE assumption with advantage ?–$negl$(n).

Game G_3: In this game, the challenger does the same as in G_2, except that it randomly choose $\boldsymbol{c}^* \leftarrow_R \mathrm{Z}_q^m$. So the signature σ^* in G_3 is independent from the choice of i_b, the probability that b'=b is exactly 1/2, namely, our VLR group signature is selfless-anonymous.

Theorem 3 (Traceability) Under the SIS assumption, our VLR group signature is traceability in the random oracle model.

Proof Assume there is an adversary A that breaks the traceability of our VLR group signature. Using A, there is an algorithm C that breaks the SIS assumption. Given a matrix $\hat{\boldsymbol{A}} \in \mathrm{Z}_q^{n\times m}$, C

tries to find a solution $\hat{\boldsymbol{x}} \in \mathrm{Z}_q^m$ such that $\|\hat{\boldsymbol{x}}\| \le poly(m)$ and $\hat{\boldsymbol{A}}\hat{\boldsymbol{x}} = \boldsymbol{0}$. The main steps are as follows:

(a) **Setup**: C randomly chooses $\boldsymbol{R} \leftarrow_R \{-1,1\}^{m\times m}$, $i^* \in \{1,2,\cdots,N\}$, $\boldsymbol{x}_{i^*,1} \leftarrow_R D_{\mathrm{Z}^m,\beta}$, $\boldsymbol{x}_{i^*,2} \leftarrow_R D_{\mathrm{Z}^m,\beta}$, and runs TrapGen$(q,n)$ to obtain a matrix $\boldsymbol{A}_{2,2} \in \mathrm{Z}_q^{n\times m}$ together with a short basis $\boldsymbol{T}_{\boldsymbol{A}_{2,2}}$ for $\Lambda_q^{\perp}(\boldsymbol{A}_{2,2})$. C sets $\boldsymbol{A}_1 = \hat{\boldsymbol{A}}, \boldsymbol{A}_{2,1} = \boldsymbol{A}_1\boldsymbol{R} - i^*\boldsymbol{A}_{2,2}$, and runs SuperSamp$(\boldsymbol{A}_1, \boldsymbol{0})$ to obtain a matrix $\boldsymbol{B} \in \mathrm{Z}_q^{n\times m}$. C computes $\boldsymbol{u}^* = \left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,1} + i^*\boldsymbol{A}_{2,2}\right] \cdot \begin{bmatrix} \boldsymbol{x}_{i^*,1} \\ \boldsymbol{x}_{i^*,2} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{x}_{i^*,1} + \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i^*,2}$, and sets $\boldsymbol{u} = \boldsymbol{u}^*$. The secret key and revocation token of member i^* is $gsk[i^*] = (\boldsymbol{x}_{i^*,1}, \boldsymbol{x}_{i^*,2})$ and $grt[i^*] = \boldsymbol{A}_1 \cdot \boldsymbol{x}_{i^*,1}$. For $i \in \{1,2,\cdots,N\}, i \ne i^*$, it chooses $\boldsymbol{x}_{i,2} \leftarrow_R D_{\mathrm{Z}^m,\beta}$, and computes $\boldsymbol{u}_i = (\boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}) \cdot \boldsymbol{x}_{i,2} = [\boldsymbol{A}_1\boldsymbol{R} + (i - i^*)\boldsymbol{A}_{2,2}]\boldsymbol{x}_{i,2}$, then gives $\boldsymbol{u}_i$ to C. Using SampleRight$(\boldsymbol{A}_1, \boldsymbol{A}_{2,2}, \boldsymbol{R}, \boldsymbol{T}_{\boldsymbol{A}_{2,2}}, \beta, \boldsymbol{u} - \boldsymbol{u}_i)$, C obtains a short vector $(\boldsymbol{x}_{i,1}, \boldsymbol{x}'_{i,2}) \in \mathfrak{C}_q^{2m}$, and returns it to i. Then, i sets its secret key $gsk[i] = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2} + \boldsymbol{x}'_{i,2})$ and revocation token $grt[i] = \boldsymbol{A}_1 \cdot \boldsymbol{x}_{i,1}$. Finally, let $Gpk = (\boldsymbol{A}_1, \boldsymbol{A}_{2,1}, \boldsymbol{A}_{2,2}, \boldsymbol{B}, \boldsymbol{u}), Gsk = (gsk[1], gsk[2], \cdots, gsk[N]), Grt = (grt[1], grt[2], \cdots, grt[N])$. Note that, by construction, the distribution of (Gpk, Gsk, Grt) is statistically close to that of the real scheme and the choice i^* is hidden from the adversary. Then, C set $RL = \varnothing$, the corruption set $U = \varnothing$, and gives (Gpk, Grt) to A.

(b) **Queries**: Adversary A can make the queries as follows:

Corruption: Request for the secret key of any member $i \in \{1,2,\cdots,N\}$. C appends i to the set U and returns $gsk[i]$.

Signing: Request for a signature on $M \in \{0,1\}^*$ of member i. C returns $\sigma = \mathrm{Sign}(Gpk, gsk[i], M)$.

(c) **Forgery**: Finally, A outputs a message M^*, a set of revocation tokens RL^* and a non-trivial forged signature $\sigma^* = (M^*, \boldsymbol{r}_i^*, \boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \boldsymbol{c}^*, \boldsymbol{d}_1^*, \boldsymbol{d}_2^*, \pi_1^*, \pi_2^*, \pi_3^*, \pi_4^*, \pi_5^*, \pi_6^*)$, satisfying the following:

1. $\mathrm{Verify}(Gpk, RL^*, \sigma^*, M^*) = \mathrm{Valid}$,
2. The (implicit) tracing algorithm fails or traces to a member outside of the coalition $U \setminus RL^*$.

Upon receiving the forged valid signature σ^*, we can apply Section 2.5 and Section 3.3 to extract $\boldsymbol{e}^*, (\boldsymbol{x}_{i,1}^*, \boldsymbol{x}_{i,2}^*)$, and $\boldsymbol{v}_i^*$ with norm at most $4\eta m^2$ by programming the random oracle twice to generate two different "Challenges" satisfying $\left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,1} + i\boldsymbol{A}_{2,2}\right] \cdot \begin{bmatrix} \boldsymbol{x}_{i,1}^* \\ \boldsymbol{x}_{i,2}^* \end{bmatrix} = \boldsymbol{u}, \boldsymbol{A}_1\boldsymbol{x}_{i,1}^* \bmod q \notin RL^*$.

Now, we consider two cases as follows:

I. If $i \ne i^*$, which happens with probability at most $\frac{N-1}{N}$, then C declares "Fail" and aborts.

II. If $i = i^*$, we note that $\boldsymbol{u}^* = \left[\boldsymbol{A}_1 \middle| \boldsymbol{A}_{2,1} + i^*\boldsymbol{A}_{2,2}\right] \cdot \begin{bmatrix} \boldsymbol{x}_{i^*,1} \\ \boldsymbol{x}_{i^*,2} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{x}_{i^*,1} + \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i^*,2}$, and $\boldsymbol{u} = \boldsymbol{u}^*$. So we have that $\boldsymbol{A}_1\boldsymbol{x}_{i^*,1} + \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i^*,2} = \boldsymbol{A}_1\boldsymbol{x}_{i,1}^* + \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i,2}^* \bmod q$. Thus, $\boldsymbol{A}_1[\boldsymbol{x}_{i,1}^* - \boldsymbol{x}_{i^*,1} + \boldsymbol{R}(\boldsymbol{x}_{i,2}^* - \boldsymbol{x}_{i^*,2})] = \boldsymbol{0} \bmod q$.

Next, we will show that, over the randomness of all algorithms, $\boldsymbol{x}_{i,1}^* - \boldsymbol{x}_{i^*,1} + \boldsymbol{R}(\boldsymbol{x}_{i,2}^* - \boldsymbol{x}_{i^*,2}) \ne \boldsymbol{0}$ with overwhelming probability.

I. If the tracing algorithm fails, we have $\mathrm{Verify}(Gpk, grt[i^*], \sigma^*, M^*) = \mathrm{Valid}$. It follows from the correctness of our VLR group signature that $\boldsymbol{A}_1\boldsymbol{x}_{i,1}^* \ne grt[i^*] = \boldsymbol{A}_1\boldsymbol{x}_{i^*,1}$. This implies that $\boldsymbol{x}_{i,1}^* \ne \boldsymbol{x}_{i^*,1}$. By $\boldsymbol{A}_1\boldsymbol{x}_{i^*,1} + \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i^*,2} = \boldsymbol{A}_1\boldsymbol{x}_{i,1}^* + \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i,2}^* \bmod q$, we can also obtain that $\boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i^*,2} \ne \boldsymbol{A}_1\boldsymbol{R}\boldsymbol{x}_{i,2}^*$, $\boldsymbol{x}_{i^*,2} \ne \boldsymbol{x}_{i,2}^*$.

II. If the tracing algorithm traces to a member $k \notin U \setminus RL^*$, namely, the following holds true:

$$\text{Verify}(Gpk, grt[k], \sigma^*, M^*) = \text{Invalid, and Verify}(Gpk, RL^*, \sigma^*, M^*) = \text{Valid.}$$

This leads to $grt[k] \in RL^*$, hence $k \notin U$. And the correctness of the revocation check implies that $\boldsymbol{A}_1 \boldsymbol{x}_{i,1}^* \bmod q = grt[k]$. We consider two cases as follows:

1. If A has never requested $gsk[i^*]$, then $(\boldsymbol{x}_{i^*,1}, \boldsymbol{x}_{i^*,2})$ is unknown to A. Because $(\boldsymbol{x}_{i^*,1}, \boldsymbol{x}_{i^*,2})$ has large min-entropy given $\boldsymbol{u}$, we have that $\boldsymbol{x}_{i,1}^* \neq \boldsymbol{x}_{i^*,1}, \boldsymbol{x}_{i,2}^* \neq \boldsymbol{x}_{i^*,2}$ with overwhelming probability.

2. If A has requested $gsk[i^*]$, then $i^* \in U$. It must be true that $i^* \neq k$, thus $grt[i^*] \neq grt[k]$. We have that $\boldsymbol{A}_1 \boldsymbol{x}_{i,1}^* \neq \boldsymbol{A}_1 \boldsymbol{x}_{i^*,1}$. As in the above, we also obtain that $\boldsymbol{A}_1 \boldsymbol{R} \boldsymbol{x}_{i^*,2} \neq \boldsymbol{A}_1 \boldsymbol{R} \boldsymbol{x}_{i,2}^*$, $\boldsymbol{x}_{i^*,2} \neq \boldsymbol{x}_{i,2}^*$.

So $\hat{\boldsymbol{x}} = \boldsymbol{x}_{i,1}^* - \boldsymbol{x}_{i^*,1} + \boldsymbol{R}(\boldsymbol{x}_{i,2}^* - \boldsymbol{x}_{i^*,2})$ is a solution of the SIS problem. By Lemma 8, we also have that $\|\hat{\boldsymbol{x}}\| \leq m^{4.5}\omega(\log^2 m)$. This concludes the proof.

5. Conclusion

In this paper, we present a simpler and efficient lattice-based VLR group signature scheme, which not only replies to the open problem proposed by Nguyen *et al*. positively, but has made a great improvement on enhancing the efficiency and saves a $O(\log n)$ factor in both sizes of the group public key and signature. We prove its security of selfless-anonymity and traceability in the random oracle model from the SIS and LWE assumption, which are as hard as worst-case lattice problems, such as SIVP_γ for some polynomial factor. However, our construction works relying on the LWE encryption schemes and only is CPA-anonymous in the random oracle model. How to construct more efficient and simple VLR group signature schemes over lattices that work without relying on encryption schemes and are with CPA-anonymity in the standard model, and even are with CCA-anonymity in the standard model should be considered in our future work.

References

[1] David Chaum and Eugène van Heyst, "Group signature," in *Proc. of International Workshop on the Theory and Application of Cryptographic Techniques*, pp. 257-265, April 8-11, 1991. Article (CrossRef Link).

[2] Dan Boneh and Hovav Shacham,"Group signatures with verifier-local revocation," in *Proc. of 11th ACM Conference on Computer and Communications Security*, pp.168-177, October 25-29, 2004. Article (CrossRef Link).

[3] R. Durán Díaz, L. Hernández Encinas and J. Muñoz Masqué, "A group signature scheme based on the integer factorization and the subgroup discrete logarithm problems," in *Proc. of 4th International Conference on Computational Intelligence in Security for Information Systems*, pp.143-150, June 8-10, 2011. Article (CrossRef Link).

[4] Benoît Libert, Thomas Peters and Moti Yung, "Scalable group signatures with revocation," in *Proc. of 31st Annual International Conference on the Theory and Application of Cryptology and Information Security*, pp.609-627, April 15-19, 2012. Article (CrossRef Link).

[5] Benoît Libert, Thomas Peters and Moti Yung, "Group signatures with almost-for free revocation," in *Proc. of 32nd Annual Cryptology Conference*, pp.571-589, August 19-23, 2012. Article (CrossRef Link).

[6] Laila El Aimani and Olivier Sanders, "Efficient group signatures in the standard model," in *Proc. of 15th International Conference on Information Security and Cryptology*, pp.410-424, November 28-30, 2013. Article (CrossRef Link).

[7] Qi Su and Wen-Min Li, "Improved Group Signature Scheme Based on Quantum Teleportation," *International Journal of Theoretical Physics*, vol. 53, no. 4, pp. 1208-1216, April, 2014. Article (CrossRef Link).
[8] Benoît Libert, Thomas Peters and Moti Yung, "Short Group Signatures via Structure Preserving Signatures: Standard Model Security from Simple Assumptions," in *Proc. of 35th Annual Annual Cryptology Conference*, pp.296-316, August 16-20, 2015. Artcle (CrossRef Link).
[9] S. Dov Gordon, Jonathan Katz and Vinod Vaikuntanathan, "A group signatures scheme from lattice assumptions," in *Proc. of 16th International Conference on the Theory and Application of Cryptology and Information Security*, pp.395-412, December 5-9, 2010. Article (CrossRef Link).
[10] Fabien Laguillaumie, Adeline Langois, Benoît Libert and Damien Stehlé, "Lattice-based group signature scheme with logarithmic signature size," in *Proc. of 19th International Conference on the Theory and Application of Cryptology and Information Security*, pp.41-61, December 1-5, 2013. Article (CrossRef Link).
[11] Adeline Langlois, San Ling, Khoa Nguyen and Huaxiong Wang, "Lattice-based group signature scheme with verifier-local revocation," in *Proc. of 17th International Conference on Practice and Theory in Public-Key Cryptography*, pp.345-361, May 26-28, 2014. Article (CrossRef Link).
[12] San Ling, Khoa Nguyen and Huaxiong Wang, "Group signature from lattices: simpler, tighter, shorter, ring-based," in *Proc. of 18th IACR International Conference on Practice and Theory in Public-Key Cryptography*, pp.427-449, March 30-April 1, 2015. Article (CrossRef Link).
[13] Phong Q. Nguyen, Jiang Zhang and Zhenfeng Zhang, "Simpler efficient group signature from lattices," in *Proc. of 18th IACR International Conference on Practice and Theory in Public-Key Cryptography*, pp.401-426, March 30-April 1, 2015. Article (CrossRef Link).
[14] Shweta Agrawal, Dan Boneh and Xavier Boyen, "Efficient lattice (H)IBE in the standard model," in *Proc. of 29th International Conference on the Theory and Application of Cryptology and Information Security*, pp.553-572, May 30-June 3, 2010. Article (CrossRef Link).
[15] Craig Gentry, Chris Peikert and Vinod Vaikuntanathan, "Trapdoor for hard lattices and new cryptographic constructions," in *Proc. of 40th ACM Symposium on Theory of Computing*, pp.197-206, May 17-20, 2008. Article (CrossRef Link).
[16] David Cash, Dennis Hofheinz, Eike Kilte and Chris Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Proc. of 29th International Conference on the Theory and Applications of Cryptographic Techniques*, pp.523-552, May 30-June 3, 2010. Article (CrossRef Link).
[17] Joel Alwen and Chris Peikert, "Generating shorter bases for hard random lattices," *International Theory of Computing Systems*, vol. 48, no. 3, pp. 535-553, April, 2011. Article (CrossRef Link).
[18] Daniele Micciancio and Oded Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM Journal on Computing Archive*, vol. 37, no. 1, pp. 267-302, April, 2007. Article (CrossRef Link).
[19] Oded Regev, "On lattice, learning with errors, random linear codes, and cryptography," in *Proc. of 37th Annual ACM Symposium on Theory of Computing*, pp.84-93, May 22-24, 2005. Article (CrossRef Link).
[20] Vadim Lyubashevsky,"Lattice signatures without trapdoors," in *Proc. of 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp.738-755, April 15-19, 2012. Article (CrossRef Link).

A The Proof of π_6 in our lattice-based VLR group signature

In this section, let $\boldsymbol{t}_0 = \boldsymbol{A}_1\boldsymbol{c} + \boldsymbol{d}_1, \boldsymbol{t}_1 = \boldsymbol{A}_1\boldsymbol{c}, \boldsymbol{y}_0 = \boldsymbol{e}, \boldsymbol{y}_1 = \boldsymbol{x}_{i,1} \in \mathrm{Z}^m$, and $\boldsymbol{y}_2 = \boldsymbol{v}_i \in \mathrm{Z}^l$. We need to generate an NIZKPoK proof (*i.e.*, π_6) for the following relation:

$$R_{com} = \{(\boldsymbol{A}_1, \boldsymbol{D}, \boldsymbol{u}, \boldsymbol{t}_0, \boldsymbol{t}_1, \eta; \boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2) \in \mathrm{Z}_q^{n\times m} \times \mathrm{Z}_q^{n\times l} \times \mathrm{Z}_q^n \times \mathrm{Z}_q^n \times \mathrm{R} \times \mathrm{Z}_q^m \times \mathrm{Z}_q^m \times \mathrm{Z}^l :$$
$$\boldsymbol{t}_0 = \boldsymbol{A}_1\boldsymbol{y}_0 - \boldsymbol{D}\boldsymbol{y}_2 + \boldsymbol{u}, \boldsymbol{t}_1 = \boldsymbol{A}_1\boldsymbol{y}_0 + \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{u}, \|\boldsymbol{y}_i\| \le \eta, i \in \{0,1,2\}\}$$

The proof is similar to [13], now, we present the basic protocol with single-bit challenge as follows:

1.The prover *P* generates a commitment Com=$(\boldsymbol{u}_k)_{k\in\{0,1\}}$, where, for each $k, \boldsymbol{u}_k \in \mathrm{Z}_q^n$ is obtained by sampling $\boldsymbol{y}_0' \leftarrow_R D_{\mathrm{Z}^m,\gamma}$, $\boldsymbol{y}_1' \leftarrow_R D_{\mathrm{Z}^m,\gamma}$, and $\boldsymbol{y}_2' \leftarrow_R D_{\mathrm{Z}^l,\gamma}$, then computing $\boldsymbol{u}_0 = \boldsymbol{A}_1\boldsymbol{y}_0' - \boldsymbol{D}\boldsymbol{y}_2' \bmod q$, and $\boldsymbol{u}_1 = \boldsymbol{A}_1\boldsymbol{y}_0' + \boldsymbol{A}_1\boldsymbol{y}_1' \bmod q$. Then, Com is sent to the verifier *V*.

2. The verifier *V* sends a challenge Chall=$c \leftarrow_R \{0,1\}$ to *P*.

3. For $i \in \{0,1,2\}$, the prover *P* does as follows:

i. Compute $\boldsymbol{z}_i = \boldsymbol{y}_i' + c \cdot \boldsymbol{y}_i$.

ii. Set $\boldsymbol{z}_i = \perp$ with probability $\zeta(\boldsymbol{z}_i, \boldsymbol{y}_i)$. Then $\boldsymbol{z} = (\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2)$ is sent to *V*.

For $i \in \{0,1,2\}$, *V* accepts it if and only if $\|\boldsymbol{z}_i\| \le 2\gamma\sqrt{m+l}$, $\boldsymbol{A}_1\boldsymbol{z}_0 - \boldsymbol{D}\boldsymbol{z}_2 = \boldsymbol{u}_0 + c \cdot (\boldsymbol{t}_0 - \boldsymbol{u})$, and that $\boldsymbol{A}_1\boldsymbol{z}_0 + \boldsymbol{A}_1\boldsymbol{z}_1 = \boldsymbol{u}_1 + c \cdot (\boldsymbol{t}_1 - \boldsymbol{u}) \bmod q$.

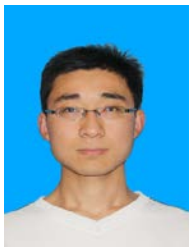
Next, we analysis the above protocol. If the prover dose as follows:

1. Sample $\boldsymbol{z}_0 \leftarrow_R D_{\mathrm{Z}^m,\gamma}, \boldsymbol{z}_1 \leftarrow_R D_{\mathrm{Z}^m,\gamma}, \boldsymbol{z}_2 \leftarrow_R D_{\mathrm{Z}^l,\gamma}$, and compute $\boldsymbol{u}_0 = \boldsymbol{A}_1\boldsymbol{z}_0 - \boldsymbol{D}\boldsymbol{z}_2 - c \cdot (\boldsymbol{t}_0 - \boldsymbol{u})$, and that $\boldsymbol{u}_1 = \boldsymbol{A}_1\boldsymbol{z}_0 + \boldsymbol{A}_1\boldsymbol{z}_2 - c \cdot (\boldsymbol{t}_1 - \boldsymbol{u})$.

2. Set $\boldsymbol{z}_i = \perp$ with probability $1 - \frac{1}{M_l}$, and output $(\boldsymbol{u}_0, \boldsymbol{u}_1, c, \boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2)$.

By lemma 2, the distribution of $(\boldsymbol{u}_0, \boldsymbol{u}_1)$ is statistically close to uniform over $\mathrm{Z}_q^n \times \mathrm{Z}_q^n$, which is the same as that in the transcript of a real proof. In addition, by the theorem 4.6 in [20], the distribution of $(\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2)$ is also statistically close to that in the transcript of a real transcript.

Similarly, applying the Fiat-Shamir transformation in a standard way, one can obtain an NIZKPoK by computing Chall=$H(\rho, \mathrm{Com})$, where $H: \{0,1\}^* \to \{0,1\}^t$ is modeled as a random oracle, and ρ represents all the other auxiliary inputs, *e.g.*, a specified message *M* to be signed.



Yanhua Zhang is currently pursuing a PhD in cryptography at Xidian University, Xi'an, China. He received his Bachelor's degree in mathematics from Taiyuan University of Science and Technology in 2012. His research interests include public key cryptography based on lattice and provable security.

Yupu Hu is a Professor and PhD supervisor at Xidian University, Xi'an, China. He received his PhD in cryptography from Xidian University in 1999, and MSc and BSc in mathematics from Xidian University in 1999 and 1987, respectively. His main research interests include Multilinear map, public key cryptography based on lattice, and fully homomorphic encryption.

Wen Gao is a PhD student at Xidian University. She received her BSc in Electronic Information Engineering from Henan University of Technology, China in 2011 and she takes a successive postgraduate and doctoral program. Her research interests include quantum attack and quantum computation.

Mingming Jiang is a Lecturer at Huaibei Normal University. He received his PhD in cryptography from Xidian University in 2014, and MSc and BSc in cryptography from Xidian University in 2010 and 2007, respectively. His research interests include public key cryptography and signature.