Achieving Strong Security and Member Registration for Lattice-based Group Signature Scheme with Verifier-local Revocation

Maharage Nisansala Sevwandi Perera[1]* and Takeshi Koshiba[2]
[1]Graduate School of Science and Engineering, Saitama University, Saitama, Japan
perera.m.n.s.119@ms.saitama-u.ac.jp
[2]Faculty of Education and Integrated Arts and Sciences, Waseda University, Tokyo, Japan
tkoshiba@waseda.jp

Abstract

A fully dynamic group signature scheme with member registration and member revocation with strong security is desirable when using group signatures in real life. Langlois, Ling, Nguyen, and Wang (PKC 2014) presented the first lattice-based group signature scheme with member revocation. Even though their scheme employs the most flexible revocation approach called Verifier-local revocation, their scheme relied on a weaker security notion and did not provide a member registration mechanism. In this paper, we obtain a fully dynamic group signature scheme by proposing a group joining protocol to their scheme. Moreover we discuss the difficulties of achieving both fully-dynamicity and strong security at the same time and provide a scheme with solutions for those difficulties.

Keywords: Lattice-Based Group Signatures, Verifier-Local Revocation, Member Registration, Dynamical-Almost-Full Anonymity

1 Introduction

In 1991 Chaum and van Heyst [9] introduced group signature schemes. In group signature schemes, any valid group member can issue signatures as a representer of the group while hiding their identity (anonymity). On the other hand, in case of dispute, the tracing authority can identify the owner of the signature (traceability). Since the group signatures were introduced, many group signature schemes have been proposed for different aspects. While some of them were focused on the efficiency of the scheme, some of them were concentrated on the security of the scheme. However, to use group signatures in practice, group signatures should be simple, efficient, and secured while being dynamic rather than static. Thus, group signature schemes should provide both member registration and revocation. New users who want to join the group should be able to register at any time. In real world, usually, the users are allowed to select their secret signing keys, and the authority validates the keys and issues member certificates. Further, group signature schemes should support member revocation. For instance, cheated members should be removed from the group and should be restricted them from signing in future. In case of static groups, the whole system has to be re-initialized once a member is registered or removed. Thus, providing a group signature scheme that dynamically facilitates both member registration and member revocation became a desirable work.

There are several approaches to realize revocation. The simplest revocation method is generating keys newly when a member is revoked [2]. Thus the group manager creates and re-distributes the group

Journal of Internet Services and Information Security (JISIS), volume: 8, number: 4 (November 2018), pp. 1-15
*Corresponding author: Graduate School of Science and Engineering, Saitama University, Saitama, Japan, Tel: +81-90-1208-6384

public key and the secret signing keys for the verifiers and the existing members. Since the keys are not generated for the revoking member, he cannot sign in future. However, producing and re-distributing keys for each member revocation is not suitable for a large group. Another approach is sending a single short public broadcast message to all members and verifiers [6]. This approach is also not convenient to use in practice as it requires to update both the members and the verifiers after each revocation. Brickell [4] proposed another revocation technique called Verifier-local Revocation (VLR), which was formalized by Boneh et al. [3] in their group signature scheme and which requires to update only the verifiers with revocation messages. Verifier-local Revocation (VLR) group signature schemes use a token system. Thus, every member of the group has a token other than their secret signing key. When a member is revoked, the group manager sets the revoking member's token to a list called *Revocation List* (*RL*) and passes RL to the verifiers. At the verification stage of a signature, the verifier validates the signature is generated on the given message, and he confirms the signer's revocation token is not in the latest RL. Since VLR requires to send the revocation information only to the verifiers who are less in number than members in practical situations, VLR seems to be the most suitable approach for any size of groups. However, most of the existing VLR group signature schemes operate in the bilinear map setting which will be insecure when the quantum computers become a reality.

In recent years, lattice-based group signatures have been an active research topic, because lattice-based cryptography considered as the most promising candidate against quantum computers. The lattice-based cryptography provides provable security under worst-case hardness assumptions. The first lattice-based group signature scheme was introduced by Gorden et al. [11] in 2010. However, their scheme faced a linear-barrier problem, i.e., the signature size and the group public key size increase with the number of members. Then, Camenisch et al. [7] proposed a scheme with an anonymous token system. However, their scheme's signature size is still linear in the number of members. Finally, Languillaumie et al. [12] suggested a scheme that overcomes the linear-barrier problem. However, none of the first three lattice-based group signature schemes are for dynamic groups. They all are for static groups. The first lattice-based group signature scheme that supports member revocation was suggested by Langlois et al. [13] in 2014. The scheme in [13] manages member revocation using Verifier-local revocation (VLR) mechanism which is the most flexible revocation approach up to now. Their scheme operates within the structure of a *Bonsai tree* of hard random lattices [8]. However, the noticeable disadvantage of this scheme is that it satisfies a weaker security notion of *selfless-anonymity*. On the other hand, since the scheme [13] facilitates only member revocation it cannot consider as a fully dynamic group signature scheme. To be a fully dynamic group signature scheme it should also satisfy member registration. Libert et al. [14] proposed a scheme based on lattices with member registration with a new tool where new users can join anonymously by contacting the group manager with their public keys. However, the scheme in [14] does not facilitate member revocation. Recently, Ling et al. [16] presented a fully dynamic group signature scheme based on lattices using accumulators, which seems to be less efficient than using VLR mechanism to manage member revocation in a larger group.

This paper aims to achieve full dynamicity for the existing VLR group signature scheme given in [13].

2 Our Contribution

The group signature scheme given in [13] provides a dynamic group signature scheme with a simple and efficient member revocation method with verifier-local revocation (VLR). However, it fails to achieve full dynamicity because it does not provide member registration. This paper delivers a new group signature scheme by adding a member registration facility to the VLR group signature scheme given in [13]. In the group signature scheme with VLR [13] all the keys are generated at the beginning and all the keys

are fixed at that time. Since they have only considered about the member revocation mechanism, they have defined all the keys at the beginning. But in case of applying member registration, keys for the members should be generated at the time of their joining. Moreover, the scheme should allow the new users to select their own secret signing keys anonymously. Thus, in this work, we suggest a group joining protocol which enables new members to choose their secret signing keys and allows the group manager to validate the secret keys of new members and issue the member certifications. In such manner, new users who want to join the group interact with the group manager via the joining-protocol. The new user selects his secret signing key and communicates with the group manager. The group manager checks the validity of the new user. If the new user is valid, then the group manager issues member certification. Here, for the member certificate generation, we use previous scheme's [13] secret signing key generation process. Thus, the revocation token of the user will be same as the revocation token used in the scheme [13]. We use a registration table called *reg* to store the member information.

The scheme in [13] has only one authority, the group manager. Our scheme has two managers because we separate authority of issuing member certification and authority of canceling the anonymity of the signatures. Thus, in our new scheme the group manager (issuer) issues member certification and revokes members while the tracing manager (opener) identifies the signers by opening the signatures. As a result, we generate two separate authority keys for the two managers in our scheme.

The previous VLR group signature scheme in [13] satisfied a weaker security notion, the selfless-anonymity. The scheme given in [15] used a stronger security notion proposed in [15] (BMW03 model). However, the scheme in [15] is for static groups. The BMW03 model facilitates two security requirements, namely, full-anonymity and full-traceability. The full-anonymity is much stronger than the selfless-anonymity. In the full-anonymity game between a challenger and an adversary, all the secret signing keys are given to the adversary. But in case of the scheme in [13], the full-anonymity cannot be applied directly for two reasons. The scheme in [13] has revocation tokens other than the secret signing keys. The revocation tokens cannot be revealed to the adversary because the adversary can identify the owner of the signature by using the revocation-token. By executing the signature verification algorithm Verify with the revocation token of any member, the adversary can confirm whether that member generated the signature or not. Moreover, since the revocation token is a part of the secret signing key in the scheme in [13], for the same reason defined above, we cannot give any secret signing keys to the adversary in [13]. Because of these reasons the scheme in [13] relies on a weaker security notion, the selfless-anonymity, which will not provide any secret information without any request and not provide any information related to the challenging indices. A security notion called *dynamical-almost-full anonymity* suggested in [19] allows to provide all the member secret signing keys to the adversary and it requires to separate the tokens from the secret signing keys (tokens should not be derived from the secret signing keys). In our scheme, since we allow the new users to select their secret signing keys, we can use the revocation token generation given in [13] without using a new revocation token generation method and achieve stronger security than the selfless-anonymity. Thus, the procedure given in [13] for key generations of members will be executed by the group manager in our scheme to generate the member certification with member revocation tokens. Thus to increase the level of the security in our scheme we can employ the dynamical-almost-full anonymity which was proposed for fully dynamic group signature schemes.

Moreover, we provide an explicit tracing algorithm to trace signers. In the previous VLR group signature scheme [13], they used an implicit tracing algorithm which requires to execute Verify in linear with the number of members until Verify returns invalid. Since it is not an efficient tracing mechanism for large groups, we provide an explicit tracing algorithm in our scheme without using the implicit tracing algorithm given in [13].

In this paper, we show how to succeed member registration for a scheme with VLR from lattices and how to achieve stronger security than the original VLR group signature scheme. Thus comparing to

Table 1: Parameters of the scheme

Parameter	Value or Asymptotic bound
Modulus q	$\omega(n^2 \log n)$
Dimension m	$\geq 2n \log q$
Gaussian parameter σ	$\omega(\sqrt{n \log q \log n})$
Integer norm bound β	$\lceil \sigma \cdot \log m \rceil$ s.t. $(4\beta+1)^2 \leq q$
Number of decomposition p	$\lfloor \log \beta \rfloor + 1$
Sequence of integers:$\beta_1, \beta_2, \beta_3, \ldots, \beta_p$	$\beta_1 = \lceil \beta/2 \rceil; \beta_2 = \lceil (\beta - \beta_1)/2 \rceil;$ $\beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil; \ldots, \beta_p = 1$
Number of protocol repetitions t	$\omega(\log n)$

the previous scheme given in [13] our scheme provides member registration, two separate managers for managing members and tracing signers, the explicit tracing mechanism, and stronger security.

3 Preliminaries

3.1 Notations

For any integer $k \geq 1$, we denote the set of integers $\{1, \ldots, k\}$ by $[k]$. We denote matrices by bold upper-case letters such as $\mathbf{A}$, and vectors by bold lower-case letters, such as $\mathbf{x}$. We assume that all vectors are in column form. The concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times k}$, is denoted by $[\mathbf{A}|\mathbf{B}] \in \mathbb{R}^{n \times (m+k)}$. The concatenation of vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^k$ is denoted by $(\mathbf{x}\|\mathbf{y}) \in \mathbb{R}^{m+k}$. If S is a finite set, $b \xleftarrow{\$} S$ means that b is chosen uniformly at random from S.

Throughout this paper we present security parameter as n and maximum number of members in a group as $N = 2^\ell \in \mathsf{poly}(n)$. The norm bound for LWE noises is b such that $q/b = \ell \tilde{\mathcal{O}}(n)$. Let χ be a b-bounded distribution over $\mathbb{Z}$. Let $k_1 := m + \ell$ and $k_2 := n + m + \ell$. We choose other parameters as in scheme [13] as given in the table 1.

Let $\mathscr{H}_1$: $\{0,1\}^* \to \mathbb{Z}_q^{n \times \ell}$, $\mathscr{H}_2$: $\{0,1\}^* \to \{1,2,3\}^t$, and $\mathscr{G}$: $\{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ be hash functions, modeled as a random oracle. Select one-time signature scheme *OTS* = (OGen, OSign, OVer), where OGen is the key generation algorithm of *OTS* key pair (**ovk**, **osk**), OSign is signature generation and OVer is signature verification functions.

3.2 Lattices

Let q be a prime and $\mathbf{B} = [\mathbf{b}_1 | \cdots | \mathbf{b}_m] \in \mathbb{Z}_q^{r \times m}$ be linearly independent vectors in $\mathbb{Z}_q^r$. The r-dimensional lattice $\Lambda(\mathbf{B})$ for $\mathbf{B}$ is defined as

$$\Lambda(\mathbf{B}) = \{\mathbf{y} \in \mathbb{Z}^r \mid \mathbf{y} \equiv \mathbf{B}\mathbf{x} \bmod q \text{ for some } \mathbf{x} \in \mathbb{Z}_q^m\},$$

which is the set of all linear combinations of columns of $\mathbf{B}$ and m is the rank of $\mathbf{B}$.

We consider a discrete Gaussian distribution for a lattice. The Gaussian function centered in a vector $\mathbf{c}$ with parameter $s > 0$ is defined as $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi \|(\mathbf{x}-\mathbf{c})/s\|^2}$ and the corresponding probability density function proportional to $\rho_{s,\mathbf{c}}$ is defined as $D_{s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/s^n$ for all $\mathbf{x} \in \mathbb{R}^n$. The discrete Gaussian distribution with respect to a lattice Λ is defined as $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = D_{s,\mathbf{c}}(\mathbf{x})/D_{s,\mathbf{c}}(\Lambda) = \rho_{s,\mathbf{c}}(\mathbf{x})/\rho_{s,\mathbf{c}}(\Lambda)$ for all

$\mathbf{x} \in \Lambda$. Since $\mathbb{Z}^m$ is also a lattice, we can define a discrete Gaussian distribution for $\mathbb{Z}^m$. By $D_{\mathbb{Z}^m,\sigma}$, we denote the discrete Gaussian distribution for $\mathbb{Z}^m$ around the origin with the standard deviation σ.

3.3 Lattice-Related Properties

The security of our scheme depends on the hardness of following lattice problems.

3.3.1 Learning With Errors (LWE)

Definition 1 ([18]). *LWE is parametrized by $n, m \geq 1, q \geq 2$, and χ. For $\boldsymbol{s} \in \mathbb{Z}_q^n$, the distribution $\mathrm{A}_{\boldsymbol{s},\chi}$ is obtained by sampling $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random and $\mathrm{e} \leftarrow \chi$, and outputting the pair $(\boldsymbol{a}, \boldsymbol{a}^T \cdot \boldsymbol{s} + \mathrm{e})$.*

There are two version of LWE problem. *Search-LWE* is to find the secret $\mathbf{s}$ and *Decision-LWE* is to distinguish LWE samples and samples chosen according to the uniformly distribution. We use the hardness of Decision-LWE problem.

For a prime power q, $b \geq \sqrt{n}\omega(\log n)$, and distribution χ, solving $LWE_{n,q,\chi}$ problem is at least as hard as solving $SIVP_\gamma$ (*Shortest Independent Vector Problem*), where $\gamma = \tilde{O}(nq/b)$ [20].

3.3.2 Short Integer Solution ($\mathbf{SIS}_{n,m,q,\beta}$)

Definition 2 ([18, 20]). *Given m uniformly random vectors $\boldsymbol{a}_i \in \mathbb{Z}_q^n$, forming the columns of a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, find a nonzero vector $\boldsymbol{x} \in \Lambda^{\perp}(\boldsymbol{A})$ such that $||\boldsymbol{x}|| \leq \beta$ and $\boldsymbol{A}\boldsymbol{x} = 0 \mod q$.*

3.3.3 Inhomogeneous Short Integer Solution ($\mathbf{ISIS}_{n,m,q,\beta}$)

Definition 3 ([13]). *Given m uniformly random vectors $\boldsymbol{a}_i \in \mathbb{Z}_q^n$, forming the columns of a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, find a vector $\boldsymbol{x} \in \Lambda_{\boldsymbol{u}}^{\perp}(\boldsymbol{A})$ such that $||\boldsymbol{x}|| \leq \beta$.*

For any m, $\beta = \mathsf{poly}(n)$, and for any $q \geq \beta \cdot \omega(\sqrt{n \log n})$, solving $SIS_{n,m,q,\beta}$ problem or $ISIS_{n,m,q,\beta}$ problem with non-negligible probability is at least as hard as solving $SIVP_\gamma$ problem, for some $\gamma = \tilde{O}(\beta\sqrt{n})$ [10].

3.4 Lattice-Related Trapdoor generation and the preimage sampling algorithms

We use a randomized nearest-plane algorithm, called, SampleD which was discussed in [10, 17] and preimage sampleable trapdoor functions (PSTFs) GenTrap and SamplePre, which were discussed in [10, 17, 1].

- SampleD($\mathbf{R}$, $\mathbf{A}$, $\mathbf{u}$, σ) outputs $\mathbf{x} \in \mathbb{Z}^m$ sampled from the distribution $D_{\mathbb{Z}^m,\sigma}$ for any vector $\mathbf{u}$ in the image of $\mathbf{A}$, a trapdoor $\mathbf{R}$ and $\sigma = \omega(\sqrt{n \log q \log n})$. The output $\mathbf{x}$ should satisfy the condition $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \mod q$.

- GenTrap(n, m, q) is an efficient randomized algorithm that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor matrix $\mathbf{R}$ for given any integers $n \geq 1, q \geq 2$, and sufficiently large $m = 2n \log q$. The distribution of the output $\mathbf{A}$ is negl(n)-far from the uniform distribution.

- SamplePre($\mathbf{A}$, $\mathbf{R}$, $\mathbf{u}$, σ) outputs a sample $\mathbf{e} \in \mathbb{Z}^m$ from a distribution that is within negligible statistical distance of $D_{\Lambda_q^{\mathbf{u}}(A),\sigma}$, on input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a trapdoor basis $\mathbf{R}$, a target image $\mathbf{u} \in \mathbb{Z}_q^n$, and the standard deviation $\sigma \geq \omega(\sqrt{\log m})$.

4 Techniques

This section first recalls the scheme given in [13] and then provides the new feature, member registration. Finally, this section shows how to achieve stronger security for the scheme with member registration and revocation.

4.1 VLR group signature scheme based on lattices

The main building block of the scheme in [13] is a Stern-like [21] interactive argument system, that allows signer to convince the verifier his validity in zero-knowledge. Their scheme operates within a Bonsai tree structure specified by a matrix $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$, and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. The signer has to prove that he is a valid member while hiding his identity d and his Bonsai signature a small vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$. For this he uses an extended vector secret signing key $\mathbf{x} = (\mathbf{x}_0||\mathbf{x}_1^0||\mathbf{x}_1^1||\ldots||\mathbf{x}_\ell^0||\mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$, where $\mathbf{x}_1^{1-d[1]},\ldots,\mathbf{x}_\ell^{1-d[\ell]}$ are zero-blocks added to the vector $\mathbf{z}$. In the scheme given in [13], group public keys is ($\mathbf{A}$, $\mathbf{u}$), member signing key is $\mathbf{gsk}[d] = \mathbf{x}^{(d)} = (\mathbf{x}_0||\mathbf{x}_1^0||\mathbf{x}_1^1||\ldots||\mathbf{x}_\ell^0||\mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$, and member revocation token is $\mathbf{grt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \in \mathbb{Z}_q^n$.
In general VLR group signature schemes consists of three algorithms.

- KeyGen(*n*,*N*): This randomized PPT algorithm takes as inputs n and N. Then it outputs a group public key **gpk**, a vector of user secret keys $\mathbf{gsk} = (\mathbf{gsk}[0],\mathbf{gsk}[1],\ldots,\mathbf{gsk}[N-1])$, and a vector of user revocation tokens $\mathbf{grt} = (\mathbf{grt}[0],\mathbf{grt}[1],\ldots,\mathbf{grt}[N-1])$, where $\mathbf{gsk}[i]$ is the i-th user's secret key and $\mathbf{grt}[i]$ is his revocation token.
- Sign(**gpk**, **gsk**[d], M): This randomized algorithm takes as inputs a secret signing key **gsk**[d], the group public key **gpk** and a message $M \in \{0,1\}^*$ and generates a group signature Σ on M.
- Verify(**gpk**, Σ, M, RL): This deterministic algorithm verifies whether the given Σ is a valid signature using the given group public key **gpk** and the message M. Then validates the signer not being revoked using RL.

Implicit Tracing Algorithm: Any VLR group signature scheme has an *implicit tracing algorithm* that takes **grt** as the secret tracing key. This algorithm can trace a signature to at least one group user who generated it. For each $i = 0,\ldots,N-1$ run Verify(**gpk**, Σ, M, RL). It outputs the index of the first user for the verification algorithm returns invalid. The tracing algorithm fails if this algorithm verifies properly for all users on the given signature. Since the implicit tracing algorithm requires to run Verify for all members, it is inappropriate for a large group.

4.2 Adding member registration

Since previous scheme in [13] only supports member revocation using VLR, it is not a fully dynamic group signature scheme. In this section, we discuss how to apply member registration feature to the previous scheme and produce a new scheme with both member registration and revocation.

We suggest a joining protocol as the solution for the member registration requirement. Here we use the techniques used in [14] to provide an efficient member registration protocol.

Joining Protocol:

Any user i having a long-term public and private key pair (**upk**[i] and **usk**[i]) samples a short vector $\mathbf{w}_i \hookleftarrow D_{\mathbb{Z}^{4m},\sigma}$ as his secret signing key, which is used to compute a syndrome $\mathbf{y}_i = \mathbf{F} \cdot \mathbf{w}_i \in \mathbb{Z}_q^{4n}$ with

public parameter $\mathbf{F}$. Then the new user i generates a signature $sig_i = \mathsf{Sign}(\mathbf{usk}[i], \mathbf{y}_i)$ and sends $(sig_i, \mathbf{y}_i)$ to the group manager. The group manager checks the validity of sig_i using $\mathbf{upk}[i]$ and checks whether $\mathbf{y}_i$ is used by previous members. If sig_i is valid and $\mathbf{y}_i$ is not used before, then he continues issuing the member certificate.

For generating the member certifications, we use the techniques used in [13] to generate members' secret keys. First, the group manager selects a fresh ℓ-bit string as the index d of the new user and then samples $\mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and computes $\mathbf{z} = \sum_{i=1}^{\ell} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]} \mod q$. Next the group manager samples $\mathbf{x}_0 \in \mathbb{Z}^m$ and defines $\mathbf{x} = (\mathbf{x}_0 || \mathbf{x}_1^0 || \mathbf{x}_1^1 || \ldots || \mathbf{x}_\ell^0 || \mathbf{x}_\ell^1)$, where $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$ are zero vectors $\mathbf{0}^m$. The revocation token of the new user is $\mathbf{grt} = \mathbf{A}_0 \cdot \mathbf{x}_0$. Finally, the group manager stores the new member's information $(d, \mathbf{y}_i, \mathbf{usk}[i], sig_i, \mathbf{x}, \mathbf{grt}, 1)$ (by inserting 1 for the new user we represent member is active) in *reg* and sends the member certificate to the new member.

We assume the new user and the group manager interact through a secured channel.

4.3 Achieving stronger security

The general VLR group signature schemes satisfy a weaker security notion called *selfless-anonymity*. However techniques in [19] provide stronger security than the selfless-anonymity called *dynamical-almost-full anonymity* for the VLR group signature schemes with both member registration and revocation. Since our scheme serves both member registration and revocation with VLR we employ the dynamical-almost-full anonymity which is defined below to secure our scheme.

The dynamical-almost-full anonymity game $\mathbf{Exp}^{anon}_{FDGS,A}(n,N)$ between a challenger and an adversary is as below.

- **Initial Phase:** The challenger C runs KeyGen to obtain a group public key **gpk**, authorities' secret keys (**ik**,**ok**). Then gives **gpk** and existing group members' secret signing keys **gsk** to the adversary A.

- **Query Phase:** The adversary A can add new users any number of times via registration query and C checks whether the new user details are already exist in the registration table *reg*. The group manager adds the new user to the group if the new user is valid and not in *reg*. Then C generates revocation token and certificate and saves new user's information in *reg*. However, C will not provide the revocation token of the newly added user to A at the time of registering. Thus, the member certification *cert* will be given without the revocation token. Moreover, A can query revocation token of any user and can access the opening oracle with any message M and a valid signature Σ.

- **Challenge Phase:** The adversary A outputs a message M^* and two distinct identities i_0, i_1. If the revocation tokens of i_0, i_1 are not revealed by A and if the challenged indices are indices of newly added users by A, then C selects a bit $b \xleftarrow{\$} \{0,1\}$, generates $\Sigma^* = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i_b], cert_{i_b}, M^*)$ and sends Σ^* to A. The adversary A still can query the opening oracle with any signature except the challenging signature. A can query revocation tokens except for challenged indices. However, A can add users to the group as before.

- **Guessing Phase:** Finally, the adversary A outputs a bit b'. If $b' = b$, then A wins.

Definition 4. *Let A be an adversary against the anonymity of a fully dynamic group signature scheme FDGS. The advantage of A in the above game is*

$$\boldsymbol{Adv}^{anon}_{FDGS,A}(n,N) = |\, Pr[\boldsymbol{Exp}^{anon}_{FDGS,A}(n,N) = 1] - 1/2 \,|\,.$$

We say that a fully dynamic group signature scheme is dynamical-almost-full anonymous if $\boldsymbol{Exp}^{anon}_{FDGS,A}$ *is negligible.*

5 New VLR group signature scheme with member registration

In this section, we first describe our new lattice-based VLR group signature scheme with member registration and revocation. Then we present the underlying interactive protocol in brief.

5.1 Description of the Scheme

Our scheme consists of two extra algorithms Join and Open than the algorithms given in [13].

Key Generation: This randomized algorithm KeyGen(n, N) creates a group public key **gpk**, the group manager key **ik**, and the tracing manager key **ok**.

1. Run GenTrap(n, m, q) to get $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T_A}$.
2. Sample $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. Sample $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for each $b \in \{0,1\}$ and $i \in [\ell]$.
4. Set the matrix $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$.
5. Run GenTrap(n,m,q) to obtain $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T_B}$.
6. Select an additional random matrix $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{4n\times 4m})$.

Finally we obtain, **gpk** = $(\mathbf{A},\mathbf{B},\mathbf{F},\mathbf{u})$, **ik** = $\mathbf{T_A}$, **ok** = $\mathbf{T_B}$.

Join: A new user i, who has a personal public and private key pair (**upk**[i], **usk**[i]) interacts with the group manager GM (issuer) to join the group, through the joining protocol.

1. User i samples a vector $\mathbf{w}_i \leftarrow D_{\mathbb{Z}^{4m},\sigma}$, and computes $\mathbf{y}_i \leftarrow \mathbf{F}\cdot\mathbf{w}_i \in \mathbb{Z}_q^{4n}$. Then he generates an ordinary digital signature $sig_i \leftarrow \mathsf{Sign}(\mathbf{usk}[i],\mathbf{y}_i)$ and sends both sig_i and $\mathbf{y}_i$, whose binary representation $\mathsf{bin}(\mathbf{y}_i)$ consists of $4n\lceil\log q\rceil = 2m$ bits to the group manager GM.
2. GM confirms $\mathbf{y}_i$ was not previously used by any member and verifies sig_i is a valid signature generated on $\mathbf{y}_i$, using $\mathsf{Vf}(\mathbf{upk}[i],\mathbf{y}_i,sig_i)$. GM aborts if any condition fails. Otherwise, GM creates a certificate for the key $cert_k = \mathsf{Sign}(\mathbf{ik},\mathbf{y}_i)$ and proceeds as follows.
   (a) Select a fresh ℓ-bit string as the index d and let $d = d[1]\ldots d[\ell] \in \{0,1\}^\ell$ be the binary representation of d.
   (b) Sample vectors $\mathbf{x}_1^{d[1]}\ldots\mathbf{x}_\ell^{d[\ell]} \leftarrow D_{\mathbb{Z}^m,\sigma}$.
   (c) Compute $\mathbf{z} = \sum_{i=1}^{\ell}\mathbf{A}_i^{d[i]}\cdot\mathbf{x}_i^{d[i]} \bmod q$.
   (d) Get $\mathbf{x}_0 \in \mathbb{Z}^m \leftarrow \mathsf{SampleD}(\mathbf{T_A},\mathbf{A}_0,\mathbf{u}-\mathbf{z},\sigma)$.
   (e) Let $\mathbf{x}_1^{1-d[1]}\ldots\mathbf{x}_\ell^{1-d[\ell]}$ be zero vectors 0^m.
   (f) Define $\mathbf{x} = (\mathbf{x}_0||\mathbf{x}_1^0||\mathbf{x}_1^1||\ldots||\mathbf{x}_\ell^0||\mathbf{x}_\ell^1) \in \Sigma^{(2\ell+1)m}$. If $||\mathbf{x}||_\infty \le \beta$ then proceed else abort.

(g) Let the revocation token of the new user i be $\mathbf{grt}[i] = \mathbf{A}_0 \cdot \mathbf{x_0}$.

Finally, GM saves the new member's details $(d, \mathbf{y}_i, \mathbf{usk}[i], sig_i, \mathbf{x}, \mathbf{grt}[i], 1)$ in *reg* and sends the member certificate $cert_i = (cert_k, d, \mathbf{x})$.

Signing : The randomized algorithm $\mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], cert_i, M)$ generates Σ on a message M, where the user i secret signing key $\mathbf{gsk}[i]=\mathbf{w}_i$.

1. Run $\mathsf{OGen}(1^n)$ to obtain a key pair (**ovk, osk**).
2. Encrypt the index d as follows. Let $\mathbf{G} = \mathscr{H}_1(\mathbf{ovk})$. Sample $\mathbf{s} \leftarrow \chi^n$, $\mathbf{e}_1 \leftarrow \chi^m$ and $\mathbf{e}_2 \leftarrow \chi^\ell$, and compute the ciphertext $(\mathbf{c}_1 = \mathbf{B}^T\mathbf{s} + \mathbf{e}_1, \mathbf{c}_2 = \mathbf{G}^T\mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor d)$.
3. Sample $\rho \xleftarrow{\$} \{0,1\}^n$, let $\mathbf{V} = \mathscr{G}(\mathbf{A}, \mathbf{u}, \mathbf{B}, M, \rho) \in \mathbb{Z}_q^{m \times n}$.
4. Compute $\mathbf{v} = \mathbf{V} \cdot (\mathbf{A}_0 \cdot \mathbf{x_0}) + \mathbf{e}_1 \mod q$ ($||\mathbf{e}_1||_\infty \leq \beta$ with overwhelming probability and $\mathbf{A}_0 \cdot \mathbf{x_0}$ is the revocation token **grt** of user i).
5. Confirm that $cert_k$ is generated on $\mathbf{y}_i$ by executing $\mathsf{Verify}(\mathbf{A}, \mathbf{y}_i, cert_k)$. Then form

$$\mathbf{P} = \left(\begin{array}{c|c} \mathbf{B}^T & \\ \mathbf{G}^T & \mathbf{I}_{m+\ell} \end{array} \right) \in \mathbb{Z}_q^{k_1 \times k_2}; \mathbf{c} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} \in \mathbb{Z}^{k_1}; \mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{k_2} \tag{1}$$

and repeat the zero knowledge interactive protocol of the commitment described in Section 5.2 $t = \omega(\log n)$ times with the public parameter ($\mathbf{A}$, $\mathbf{F}$, $\mathbf{u}$, $\mathbf{V}$, $\mathbf{v}$, $\mathbf{P}$, $\mathbf{c}$) and prover's witness $(\mathbf{x}, \mathbf{e}_1, \mathbf{e})$ to make the soundness error negligible and prove that user is certified. Then make it non-interactive using the Fiat-Shamir heuristic as a triple, $\Pi = (\{CMT^{(k)}\}_{k=1}^t, CH, \{RSP^{(k)}\}_{k=1}^t)$, where CH = $(\{Ch^{(k)}\}_{k=1}^t) = \mathscr{H}_2(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$.

6. Compute $OTS; sig = \mathsf{OSig}(\mathbf{osk}, (\mathbf{c}_1, \mathbf{c}_2, \Pi))$.
7. Output signature $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$.

Verification : $\mathsf{Verify}(\mathbf{gpk}, M, \Sigma, RL = \{\{\mathbf{u}_i\}_i\})$ checks whether the given Σ is valid on the given M and signer is a valid member as follows.

1. Parse Σ as $(\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$, and get $\mathbf{V} = \mathscr{G}(\mathbf{A}, \mathbf{u}, \mathbf{B}, M, \rho) \in \mathbb{Z}_q^{m \times n}$.
2. If $\mathsf{OVer}(\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig) = 0$ then return 0.
3. Parse Π as $(\{CMT^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{RSP^{(k)}\}_{k=1}^t)$.
4. If $(Ch^{(1)}, \ldots, Ch^{(t)}) \neq \mathscr{H}_2(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$ return 0 else proceed.
5. Form $\mathbf{P}$, $\mathbf{c}$ as in (1) and for $k = 1$ to t run the verification steps of the commitment scheme to validate $RSP^{(k)}$ with respect to $CMT^{(k)}$ and $Ch^{(k)}$. If any of the conditions fails then output invalid and hold.
6. For each $\mathbf{u}_i \in RL$ compute $\mathbf{e}_i^{'} = \mathbf{v} - \mathbf{V} \cdot \mathbf{u}_i \mod q$ to check whether there exists an index i such that $||\mathbf{e}_i^{'}||_\infty \leq \beta$. If so return invalid.

7. Return valid.

Open : Open(**ok**, M, Σ, *reg*) functions as follows, where **ok** = $\mathbf{T_B}$.

1. Let $\mathbf{G} = \mathscr{H}_1(\mathbf{ovk})$.
2. Then for $i \in [\ell]$, sample $\mathbf{y}_i \leftarrow \mathsf{SamplePre}(\mathbf{T_B}, \mathbf{B}, \mathbf{g}_i, \sigma)$.
3. Let $\mathbf{Y} = [\mathbf{y}_1 | \ldots | \mathbf{y}_\ell] \in \mathbb{Z}^{m \times \ell}$, where $\mathbf{B} \cdot \mathbf{Y} = \mathbf{G}$.
4. Compute $d' = (d'_1, \ldots, d'_\ell) = \mathbf{c}_2 - \mathbf{Y}^T \cdot \mathbf{c}_1 \in \mathbb{Z}_q^\ell$.
5. For each $i \in [\ell]$, if d'_1 is closer to 0 than to $\lfloor q/2 \rfloor$ modulos q, then let $d_i = 0$. Otherwise, let $d_i = 1$.
6. Create $d = (d'_1, \ldots, d_\ell) \in \{0,1\}^\ell$ and return d.

5.2 The Underlying ZKAoK for the Group Signature Scheme

The Stern-like [21] interactive system allows the signer to convince the verifier, he is a certified and valid group member who followed the signature generation correctly. The public parameters consists of matrices $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \ldots | \mathbf{A}_\ell^0 | \mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, $\mathbf{F} \in \mathbb{Z}_q^{4n \times 4m}$, $\mathbf{V} \in \mathbb{Z}_q^{m \times n}$, and $\mathbf{P} \in \mathbb{Z}_q^{k_1 \times k_2}$, and vectors $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{v} \in \mathbb{Z}_q^n$, $\mathbf{c} \in \mathbb{Z}^{k_2}$. The prover's inputs are the vectors $\mathbf{x} = (\mathbf{x}_0 || \mathbf{x}_1^0 || \mathbf{x}_1^1 || \ldots || \mathbf{x}_\ell^0 || \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$, $\mathbf{e}_1 \leftarrow \chi^m$, $\mathbf{w} \in [-\beta, \beta]^{4m}$, $\mathbf{y} \in \{0,1\}^{2m}$, and $\mathbf{e} \in \mathbb{Z}^{k_2}$. The prover's goal is to convince the verifier the following four statements.
1. $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \mod q$ and $\mathbf{x} \in \mathsf{Secret}_\beta(d)$.
2. $||\mathbf{e}_1||_\infty \leq \beta$ and $\mathbf{V} \cdot (\mathbf{A}_0 \cdot \mathbf{x}_0) + \mathbf{e}_1 = \mathbf{v} \mod q$.
3. $\mathbf{F} \cdot \mathbf{w} = \mathbf{H}_{4n \times 2m} \cdot \mathbf{y} \mod q$.
4. $\mathbf{Pe} + (0^{k_1 - \ell} || \lfloor q/2 \rfloor d) = \mathbf{c} \mod q$.

To prove the goal 1 and 2 we can directly use the interactive protocol given in [13]. We can use the proof provided in [14] for the goal 3 and the proof given in [15] for the goal 4. We can combined all the proofs together and use as the interactive protocol for our scheme.

6 Correctness and Security Analysis of the Scheme

In this section we show the correctness and the security of our scheme. First, we provide the correctness of the proposed scheme. Then we provide the anonymity of our scheme under the hardness of LWE problem and the traceability and non-frameability under the hardness of SIS problem.

6.1 Correctness

For all **gpk**, **gsk**, and **grt**,

1. $\mathsf{Verify}(\mathbf{gpk}, M, \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], cert_i, M), RL) = \mathsf{Valid}$ and $\mathbf{grt}[i] \notin RL$.
2. $\mathsf{Open}(\mathbf{gpk}, \mathbf{ok}, M, \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], cert_i, M), reg) = i$

6.2 Anonymity

Theorem 1. *In the random oracle model, the proposed scheme is dynamical-almost-full anonymous based on the hardness of $LWE_{n,q,\chi}$.*

We use a sequence of games to prove our scheme is dynamical-almost-full anonymous.

Game 0: The challenger C obtains the group public key and authorities' keys by executing KeyGen$(1^n, 1^N)$. Then C gives **gpk** and all the existing group members' secret keys **gsk** to the adversary A. In the query phase, A can add new users to the group through the registration query. In the registration query, C will accept valid members but will provide the certification $cert = (accept, d, \varepsilon)$ without the revocation token information. Later, A can request for revocation tokens of any member, and can access opening query for any signature. In the challenge phase, first, A sends two indices (i_0, i_1) and a message M^*. Then C checks whether (i_0, i_1) are newly added by the adversary and not used to query revocation tokens. Then C generates the challenging signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ for a random bit $b \leftarrow \{0,1\}$ and sends to A. The adversary A's goal is to guess b' the index is used to generate the challenging signature. If $b' = b$ then the experiment returns 1. Otherwise returns 0.

Game 1: In this game, the challenger C slightly modifies Game 0. Here C produces *OTS* pair $(\mathbf{ovk}^*, \mathbf{osk}^*)$ at the beginning of the game. If A accesses the opening oracle with a valid signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$, where $\mathbf{ovk} = \mathbf{ovk}^*$, C returns a random bit and aborts the game. However, A accessing opening oracle with a signature Σ, where $\mathbf{ovk} = \mathbf{ovk}^*$ contradicts the strong unforgeability of *OTS*, and since $\mathbf{ovk}^*$ is independent of A's view, probability of A comes up with $\mathbf{ovk} = \mathbf{ovk}^*$ is negligible. Even after generating the challenging signature if A comes up with a valid signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$, where $\mathbf{ovk}=\mathbf{ovk}^*$, then sig is a forged one-time signature, which violates the strong unforgeability of *OTS*. Thus, without lose of generality we assume that A does not request for opening of a valid signature, where $\mathbf{ovk} = \mathbf{ovk}^*$ and C aborting the game is negligible.

Game 2: In this game, C modifies the generation of encrypting matrices $\mathbf{B}$ and $\mathbf{G}$ and programs the random oracle $\mathscr{H}_1$ accordingly. At the beginning of the game, C chooses uniformly random matrices $\mathbf{B}^* \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{G}^* \in \mathbb{Z}_q^{n \times \ell}$ and sets $\mathscr{H}_1(\mathbf{ovk}^*)=\mathbf{G}^*$. To answer the opening oracle requests of A with $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$, C samples a matrix $\mathbf{Y} \leftarrow (D_{z^m,\sigma})^\ell$, and computes $\mathbf{G} = \mathbf{B}^* \cdot \mathbf{Y}$. This $\mathbf{G}$ is used to answer the signature openings in later and keep track of ($\mathbf{ovk}$, $\mathbf{Y}$, $\mathbf{G}$) to be reused if A repeats the same requests for $\mathscr{H}_1(\mathbf{ovk})$. For the view of A, the distribution of $\mathbf{G}^*$ is statistically close to the real experiment [10]. In this way, Game 2 is indistinguishable from Game 1.

Game 3: In this game, without honestly generating the legitimate non-interactive proof Π, C simulates the proof without using the witness. C invokes the simulator for each $k \in [t]$ and then programs the random oracle $\mathscr{H}_2$ accordingly. The challenging signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ is statistically close to Σ^* in the previous game since the argument system is statistically zero-knowledge. Thus, Game 3 is indistinguishable from Game 2.

Game 4: In this game, C replaces the original revocation token by a vector sampled uniformly random. The original game has $\mathbf{v} = \mathbf{V} \cdot \mathbf{grt}[i_b] + \mathbf{e}_1 \mod q$. Here C samples a vector $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$ uniformly and computes $\mathbf{v} = \mathbf{V} \cdot \mathbf{t} + \mathbf{e}_1 \mod q$, where $\mathbf{V}$ is uniformly random over $\mathbb{Z}_q^{m \times n}$, $\mathbf{e}_1$ is sampled from the error distribution χ. Since C replaces only the revocation token $\mathbf{grt}[i_b]$ with $\mathbf{t}$ and the rest of the game is same as Game 3, Game 4 is indistinguishable from Game 3.

Game 5: In this game the challenger C makes $\mathbf{v}$ truly uniform by sampling $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$. In such away, C makes revocation token totally independent of the bit b. C samples $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ and sets $\mathbf{v} = \mathbf{y}$. In Game 4, the pair ($\mathbf{V}$, $\mathbf{v}$) is a proper $LWE_{n,q,\chi}$ instance. Under the assumption of the $LWE_{n,q,\chi}$ problem is hard, Game 4 and Game 5 are indistinguishable.

Game 6: In this game C modifies the generation of ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*)$ in the challenge phase. Let $\mathbf{c}_1^* = \mathbf{z}_1$ and $\mathbf{c}_2^* = \mathbf{z}_2 + \lfloor q/2 \rfloor d_b$, where $\mathbf{z}_1 \in \mathbb{Z}^m$ and $\mathbf{z}_2 \in \mathbb{Z}^{2m}$ are uniformly random. d_b is the index of the adversary's challenging bit. The rest of the game is same as Game 5. Game 5 and Game 6 are indistinguishable under the assumption of the hardness of $LWE_{n,q,\chi}$. Indeed, if A can distinguish two games, then he can also solve the LWE problem.

Game 7: Finally, C makes Σ^* totally independent of the challenging bit b. C samples $\mathbf{z}_1' \in \mathbb{Z}_q^m$ and $\mathbf{z}_2' \in \mathbb{Z}_q^{2m}$ uniformly random and sets $\mathbf{c}_1^* = \mathbf{z}_1'$ and $\mathbf{c}_2^* = \mathbf{z}_2'$. Thus, Game 6 and Game 7 are statistically indistinguishable. Game 7 is totally independent from the challenger's bit b. Thus, the advantage of the adversary in this game is 0.

Hence, these games prove that our scheme is secure with dynamical-almost-full anonymity.

6.3 Traceability

Theorem 2. *Based on the hardness of SIS problem, the proposed scheme is traceable, in the random oracle model.*

The adversary A wins traceability game if he can generate a valid signature either traces to an inactive user or cannot be traced to a user.

We constructs a PPT algorithm B that solves SIS problem with non-negligible probability. The adversary A, who has **gpk** and **ok** outputs (M, Σ) in the traceability game. B interacts with A by answering for the queries of A. A can add new users and replace members' personal public keys. Moreover, he can query for secret signing keys and revocation tokens of any member.

Finally, A outputs a message M^*, set of revocation tokens RL^*, and a forgery signature Σ^*=$(\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ on message M^*, such that $\mathsf{Verify}(\mathbf{gpk}, M^*, \Sigma^*, RL^*) = \mathsf{Valid}$. B opens Σ^* and obtains the index. The improved Forking Lemma [5] guarantees that, with probability at least $1/2$, B can obtain 3-fork involving tuple $(M^*, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$ running A up to $32 \cdot Q_H/(\varepsilon - 3^{-t})$ times with the same tape. Rest of the proof flows as in [13] and we can extract vectors $\mathbf{y} = (\mathbf{y}_0 || \mathbf{y}_1^0 || \mathbf{y}_1^1 || \ldots || \mathbf{y}_\ell^0 || \mathbf{y}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$, $\mathbf{e}_1^* \in \mathbb{Z}^m$, $\mathbf{e}^* \in \mathbb{Z}^m$, and $\mathbf{w}^* \in \{0,1\}^m$ from the proof of knowledge Π^* such that,

1. $\mathbf{y} \in \mathsf{Secret}_\beta$ for some $d \in \{0,1\}^\ell$, and $\mathbf{A} \cdot \mathbf{y} = \mathbf{u} \mod q$.
2. $||\mathbf{e}_1^*||_\infty \leq \beta$ and $\mathbf{v}^* = \mathbf{V}^* \cdot (\mathbf{A}_0 \cdot \mathbf{y}_0) + \mathbf{e}_1^* \mod q$.
3. $||\mathbf{e}^*||_\infty \leq b$ and $(\mathbf{B}^T \mathbf{s}^* + \mathbf{e}_1^*) = \mathbf{c}_1 \mod q, (\mathbf{G}^T \mathbf{s}^* + \mathbf{e}_2^* + \lfloor q/2 \rfloor d^*) = \mathbf{c}_2 \mod q$.
4. $\mathbf{F} \cdot \mathbf{w}^* = \mathbf{H}_{4n \times 2m} \cdot \mathbf{h} \mod q$.

The zero-knowledge interactive protocol system guarantees that $\mathbf{A}_0 \cdot \mathbf{y}_0$ not in RL^*, and also Verify confirms this condition. Thus, the forgery signature tracing to an inactive user is negligible. Moreover, interactive protocol system guarantees again, when signing the user has to prove the validity with his witness and needs to encrypt the index.

This concludes our proof of traceability.

6.4 Non-frameability

Theorem 3. *Based on the hardness of SIS problem, the proposed scheme is non-frameable, in the random oracle model.*

Suppose there is a frameable adversary A with advantage ε, who creates a forgery (M^*, Σ^*) that opens to an innocent user i^* who did not sign M^*. We construct a **PPT** algorithm B that solves $SIS_{4n,4m,q,\beta''}$ problem by taking $\bar{A} \in \mathbb{Z}_q^{4n \times 4m}$ and finds a non-zero short vector $\mathbf{w} \in \Lambda_q^{\perp}(\bar{A})$.

B generates all the public keys and authorities' keys honestly. Then B interacts with A by sending group public key **gpk** and authorities' keys (**ik**, **ok**). B responses to A's all queries. A can act as a corrupted group manager and add a new user i to the group. When A requests user i to generate a signature on a message M, B generates and returns the signature Σ=($\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}$).

Finally, A outputs Σ^*=($\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*$) signed on a message M^* and which opens to i^* who did not sign the message. Thus, (M^*, Σ^*) should frame user i^*. B has a short vector $\mathbf{z}_{i^*} = \mathbf{F} \cdot \mathbf{x}_{i^*} \mod q$. To solve SIS instance B should have another short vector $\mathbf{z}_{i'} = \mathbf{F} \cdot \mathbf{x}_{i'} \mod q$. To compute such a vector, B proceeds by running A sufficient times and applying Improved Forking Lemma [5]. From the corresponding responses of Π^*, B can extract a short vector $\mathbf{x}_{i'}$, where $\mathbf{z}_{i^*} = \mathbf{F} \cdot \mathbf{x}_{i'} \mod q$. According to the Stern-like proof of knowledge, with overwhelming probability, we say $\mathbf{x}_{i'} \neq \mathbf{x}_{i^*}$. The difference $\mathbf{h} = \mathbf{x}_{i'} - \mathbf{x}_{i^*}$ is a suitable short non-zero vector, which is a solution for SIS problem.

This proves the non-frameability of the proposed scheme.

7 Conclusion

This work focuses on facilitating member registration mechanism for the scheme given in [13]. Thus we suggest a joining protocol to the existing VLR lattice-based group signature scheme [13]. As a result, this work provides a new scheme based on lattices with both member registration and member revocation with VLR using the scheme given in [13]. The proposed scheme consists of a joining-protocol with member revocation token generation, encryption of the user index that requires for the explicit tracing mechanism. Moreover, we make the new scheme stronger in security than the selfless-anonymity by employing the dynamical-almost-full anonymity.

Acknowledgments

This work is supported in part by JSPS Grant-in-Aids for Scientic Research (A) JP16H01705 and for Scientic Research (B) JP17H01695.

References

[1] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In *Proc. of the 15th IACR International Conference on Practice and Theory of Public-Key Cryptography (PKC'12), Darmstadt, Germany*, volume 7293 of *Lecture Notes in Computer Science*, pages 280–297. Springer, Berlin, Heidelberg, May 2012.

[2] G. Ateniese, D. Song, and G. Tsudik. Quasi-efficient revocation of group signatures. In *Financial Cryptography, 6th International Conference on Financial Cryptography (FC'02), March 2002, Revised Papers*, volume 2357 of *Lecture Notes in Computer Science*, pages 183–197. Springer-Verlag Berlin Heidelberg, 2003.

[3] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *Proc. of the 11th ACM conference on Computer and Communications Security (CCS'04), Washington DC, USA*, pages 168–177. ACM, October 2004.

[4] E. Brickell. An efficient protocol for anonymously providing assurance of the container of the private key. *Submitted to the Trusted Comp. Group (April 2003)*. 2003.

[5] E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *Proc. of the 3rd IACR International Workshop on Practice and Theory of Public-Key*

Cryptography (PKC'00), Melbourne, Australia, volume 1751 of *Lecture Notes in Computer Science*, pages 276–292. Springer, Berlin, Heidelberg, January 2000.

[6] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Proc. of the 22nd Annual International Cryptology Conference (CRYPTO'02), Santa Barbara, California, USA*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, Berlin, Heidelberg, August 2002.

[7] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *Proc. of the 8th International Conference (SCN'12), Amalfi, Italy*, volume 7485 of *Lecture Notes in Computer Science*, pages 57–75. Springer, Berlin, Heidelberg, September 2012.

[8] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10), French Riviera*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, Berlin, Heidelberg, May–June 2010.

[9] D. Chaum and E. Van Heyst. Group signatures. In *Proc. of the 1991 International Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'91), Brighton, UK*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, Berlin, Heidelberg, April 1991.

[10] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of the 40th Annual ACM Symposium on Theory of Computing (STOC'08), Victoria, British Columbia, Canada*, pages 197–206. ACM, May 2008.

[11] S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Proc. of the 16th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'10), Singapore*, volume 6477 of *Lecture Notes in Computer Science*, pages 395–412. Springer, Berlin, Heidelberg, December 2010.

[12] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *Proc. of the 19th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'13), Bengaluru, India, Part 2*, volume 8270 of *Lecture Notes in Computer Science*, pages 41–61. Springer, Berlin, Heidelberg, December 2013.

[13] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *Proc. of the 17th International Conference on Practice and Theory in Public-Key Cryptography (PKC'14), Buenos Aires, Argentina*, volume 8383 of *Lecture Notes in Computer Science*, pages 345–361. Springer, Berlin, Heidelberg, March 2014.

[14] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *Proc. of the 22nd International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'16), Hanoi, Vietnam, Part 2*, volume 10032 of *Lecture Notes in Computer Science*, pages 373–403. Springer, Berlin, Heidelberg, December 2016.

[15] S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: simpler, tighter, shorter, ring-based. In *Proc. of the 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg (PKC'5), MD, USA*, volume 9020 of *Lecture Notes in Computer Science*, pages 427–449. Springer, Berlin, Heidelberg, March-April 2015.

[16] S. Ling, K. Nguyen, H. Wang, and Y. Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In *Proc. of the 15th International Conference on Applied Cryptography and Network Security (ACNS'17), Kanazawa, Japan*, volume 10355 of *Lecture Notes in Computer Science*, pages 293–312. Springer, Cham, July 2017.

[17] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'12), Cambridge, UK*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, Berlin, Heidelberg, April 2012.

[18] C. Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.

[19] M. N. S. Perera and T. Koshiba. Achieving almost-full security for lattice-based fully dynamic group sig-

natures with verifier-local revocation. In *Proc. of the 14th International Conference on Information Security Practice and Experience (ISPEC'8), Tokyo, Japan*, volume 11125 of *Lecture Notes in Computer Science*, pages 229–247. Springer, Cham, September 2018.

[20] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of the 37th annual ACM symposium on Theory of computing (STOC'05), Baltimore, MD, USA*, pages 84–93. ACM, May 2005.

[21] J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

Author Biography

Maharage Nisansala Sevwandi Perera received the B.Sc. degree from Rajarata University of Sri Lanka, M.Sc. degree in Computer science from University of Peradeniya, Sri Lanka, and Ph.D. degree from Saitama University, Japan.

Takeshi Koshiba received the PhD degree from Tokyo Institute of Technology. He is a full professor at the Department of Mathematics, the Faculty of Education and Integrated Arts and Sciences, Waseda University. His interests include theoretical and applied cryptography, the randomness in algorithms, and quantum computing and cryptography.