Achieving Strong Security and Verifier-local Revocation for Dynamic Group Signatures from Lattice Assumptions

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. Both member registration and member revocation are essential features in group signature schemes. In ASIACRYPT 2016 Libert, Ling, Mouhartem, Nguyen, and Wang suggested a simple joining mechanism with their lattice-based group signature scheme with member registration. However, their scheme does not support member revocation. Verifier-local revocation is a member revocation approach in group signature schemes, which only requires the verifiers to keep the revocation messages while existing members have no burden. Since there is no workload for existing members related to revocation messages, verifier-local revocation method became the most suitable revocation approach for any environment. However, original group signature schemes with verifier-local revocability satisfy weaker security. This paper adds verifier-local revocation mechanism to the Libert's (ASIACRYPT 2016) scheme to produce a fully dynamic lattice-based group signature scheme with member registration and member revocation using verifier-local revocation mechanism. Moreover, the resulted scheme achieves stronger security than the security in the original group signature schemes with verifier-local revocation.

Keywords: lattice-based group signatures, verifier-local revocation, almost-full anonymity, dynamical-almost-full anonymity, member registration

1 Introduction

Group Signature schemes introduced by Chaum and van Heyst [14] enable group members to sign messages on behalf of the group while hiding their identity (anonymity). However, in case of dispute, the tracing authority can cancel the anonymity of signatures to identify the signer (traceability). These two features, anonymity and traceability allow group signatures to find applications in real-life. For instance, e-commerce systems, road-to-vehicle communication systems, and key-card access. In a theoretical manner, forming a secure and efficient group signature scheme that facilitates both member registration and revocation is both interesting and challenging task. Bellare et al. [4] (BMW03 model) proposed two formal and strong security notions called *full-anonymity* and *full-traceability* for static group signatures. Then Bellare et al. [5] used the BMW03 model to present a dynamic group signature scheme which supports only member registration. Recently, Bootel et al. [8] provided a security definition for fully dynamic group signatures.

In recent years, lattice-based group signatures have been an active research topic because lattice-based cryptography provides provable security under worst-case hardness assumptions. Gorden et al. [16] proposed the first lattice-based group signature scheme in 2010. However, the first lattice-based group signature scheme that supports member revocation was suggested by Langlois et al. [19]. The scheme in [19] manages member revocation using Verifier-local revocation (VLR) mechanism. On the other hand, the scheme presented by Libert et al. [20] provides member registration. The scheme in [20] provides a simple joining mechanism with zero-knowledge argument system that allows the valid signers to proof that their secret key is certified by the group manager. However, the scheme in [20] does not support member revocation. Thus the scheme in [20] is not fully dynamic. Ling et al. [22] presented a fully dynamic group signature scheme based on lattices using accumulators. Verifier-local revocation (VLR) mechanism is efficient than using accumulators. Especially, when considering large groups, VLR is more suitable than accumulators.

We focus on applying membership revocation facility using VLR to the scheme given in [20]. There are several revocation approaches. The simplest revocation method is that the group manager generates the group public key and secret keys of all members newly except for the revoked member and re-distributes the keys [3]. However, this is not suitable for large groups. Another approach is broadcasting a small public membership message to all signers and verifiers, as in [6, 12]. However, still, signers have to obtain revocation details at the time of signing. On the other hand, Verifier-local Revocation (VLR) sends revocation messages only to the verifiers. Since the number of verifiers is less than the number of signers, VLR method seems to be the most suitable approach for any size of groups.

Verifier-local Revocation (VLR) was proposed by Brickell [10] and formalized by Boneh et al. [7] in their group signature scheme. The *Verifier-local Revocation* (VLR) group signature scheme uses a token system, and when a member is revoked, the revoking member's token is added to a list called *Revocation List* (RL). Thus the verifier uses RL to authenticate the signer at the signature verification stage. In such manner, in VLR group signature schemes, the verifiers do "signature-check" and "revocation-check". Since VLR does not require to generate keys newly or keep track of revocation information with the existing members, it is more convenient than any other approach. When a member is revoked, VLR only asks to send the revocation information to the verifiers. Thus, VLR is suitable for any size of groups.

When applying VLR to an existing scheme, we have to focus on several problems. Since the original VLR group signature scheme based on lattices [19] relied on a weaker security notion called *selfless-anonymity*, when VLR is suggesting to an existing scheme, the existing scheme's security becomes weaker. Thus, if we want to make the resulting scheme's security stronger, then we have to consider a security notion like *almost-full anonymity* [25], which is for partially dynamic VLR schemes or *dynamical-almost-full anonymity* [24], which is for fully dynamic VLR schemes. Moreover, in general VLR schemes, revocation tokens are generated as a part of the secret signing key. But, when we apply the almost-full anonymity or the dynamical-almost-full anonymity, we have to separate generation of member revocation token from the secret signing key. Thus, we have to concern about the member revocation token generation without affecting the construction of the existing scheme.

This paper aims to achieve fully dynamic group signature scheme with strong security by proposing VLR technique to an existing member registration scheme with ease.

1.1 Our Contribution

This paper proposes a scheme by applying VLR revocation mechanism to the scheme given in [20]. The group signature scheme with VLR [19] uses revocation token, which is a part of the secret signing key. However, in case of the full-anonymity game, which is described in the BMW03 model, the adversary is given all the members secret signing keys. Hence, for the group signatures with VLR, achieving full-anonymity is technically difficult since we cannot give both the secret signing keys and the revocation tokens to the adversary at the anonymity game. If the revocation tokens are given to the adversary, he can execute the verification algorithm Verify with the revocation tokens of the challenged indices and identify the index which is used to generate the challenging signature. The scheme in [25] suggested a new security notion called almost-full anonymity, which does not provide any revocation tokens unless requested by the adversary and which does not generate the challenging signature for the indices whose revocation tokens are queried. The scheme in [24] presented a security notion called dynamical-almost-full anonymity, which is an extended version of the almost-full anonymity for fully dynamic group signature schemes. Moreover, if the secret signing keys are given to the adversary, then he can generate the revocation tokens of the challenged indices using the secret signing keys and execute Verify to identify the index, which is used to create the challenging signature. Thus, we use dynamical-almost-full anonymity to secure our scheme, and we use a vector related to the secret signing key (but not the part of the secret signing key) as the revocation token. However, since the group manager should know the revocation token, we select a vector which is generated by the group manager. Otherwise, a cheating member can present a fake revocation token to the group manager at the time of revoking or to the verifier at the time of signing.

This paper highlights the difficulties of achieving strong security while providing member revocation with VLR. Moreover, this paper shows how to accomplish strong security and member revocation with VLR without affecting the structure of the existing scheme in [20].

2 Preliminaries

2.1 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 $\mathbf{B} \in \mathbb{R}^{n \times k}$ denoted by $[\mathbf{A}|\mathbf{B}] \in \mathbb{R}^{n \times (m+k)}$. The concatenation of vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^k$ denoted by $(\mathbf{x}\|\mathbf{y}) \in \mathbb{R}^{m+k}$. The Euclidean norm of $\mathbf{x}$ is denoted by $||\mathbf{x}||$ and the infinity norm is denoted by $||\mathbf{x}||_\infty$. The Euclidean norm of matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ with columns $(\mathbf{b}_i)_{i \leq n}$ is denoted by $||\mathbf{B}|| = \max_{i \leq n} ||\mathbf{b}_i||$. If $\mathbf{B}$ is a full column-rank, then its Gram-Schmidt marginalization is denoted by $\tilde{\mathbf{B}}$. If S is a finite set, the uniform distribution over S is denoted by $U(S)$. The action of sampling x according to the uniform distribution is denoted by $x \hookleftarrow U(S)$.

Throughout this paper, we present the security parameter as $\lambda > 0$ and the maximum number of members in a group as $N = 2^\ell \in \mathsf{poly}(\lambda)$. Then choose lattice parameter $n = \mathcal{O}(\lambda)$, prime modulus $q = \tilde{\mathcal{O}}(\ell n^3)$, dimension $m = 2n\lceil \log q \rceil$, Gaussian parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$, infinity norm bounds $\beta = \sigma\omega(\log m)$ and $b = \sqrt{n}\omega(\log n)$. Choose a hash function $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$ for some $t = \omega(\log n)$, which will be modeled as a random oracle in the proof of security. Let χ be a b-bounded distribution over $\mathbb{Z}$.

2.2 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 $\mathbf{B}$. The value m is the rank of $\mathbf{B}$.

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

2.3 Lattice-Related Computational Problems

The security of our scheme relies on the hardness of the two lattice-based problems defined below.

Learning With Errors (LWE)

Definition 1. *Learning With Errors (LWE) [23] is parametrized by integers $n, m \geq 1$, and $q \geq 2$. For a vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ and χ, the distribution $\mathrm{A}_{\boldsymbol{s},\chi}$ is obtained by sampling $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random and choosing* $\mathrm{e} \leftarrow \chi$*, and outputting the pair* $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e)$.

There are two LWE problems: Search-LWE and Decision-LWE. While Search-LWE is to find the secret $\mathbf{s}$ given LWE samples, 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 $\gamma = \tilde{O}(nq/b)$ [27].

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

Definition 2. *Short Integer Solution ($SIS_{n,m,q,\beta}$ [23, 27]) is as follows. 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 $\boldsymbol{x} \in \mathbb{Z}^m$ such that $||\boldsymbol{x}|| \leq \beta$ and $\boldsymbol{A}\boldsymbol{x} = 0 \mod q$.*

For any m, $\beta = \mathsf{poly}(n)$, and for any $q \geq \sqrt{n}\beta$, solving $SIS_{n,m,q,\beta}$ problem with non-negligible probability is at least as hard as solving $SIVP_\gamma$ problem, for some $\gamma = \tilde{O}(\beta\sqrt{n})$ [15].

2.4 Lattice-Related Algorithms

Lemma 1 ([9, Lemma. 2.3]). *$\mathsf{GPVSample}$ is a PPT (probabilistic polynomial-time) algorithm that takes a basis $\boldsymbol{B}$ of a lattice $\Lambda \subseteq \mathbb{Z}^n$ and $s \geq ||\widetilde{\boldsymbol{B}}||.\Omega(\sqrt{\log n})$ as inputs, and outputs vectors $\boldsymbol{b} \in \Lambda$ with distribution $D_{\Lambda,s}$.*

Lemma 2 ([2, Theorem. 3.2]). *$\mathsf{TrapGen}$ is a PPT algorithm that takes $1^n, 1^m$ and an integer $q \geq 2$, where $m \geq \Omega(n \log q)$ as inputs, and outputs a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and a basis $\boldsymbol{T_A}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$. The distribution of the output $\boldsymbol{A}$ is within statistical distance $2^{-\Omega(n)}$ to $U(\mathbb{Z}_q^{n\times m})$, and $||\widetilde{\boldsymbol{T_A}}|| \leq \mathcal{O}(\sqrt{n \log q})$.*

Lemma 3 ([13, Lemma. 3.2]). *$\mathsf{ExtBasis}$ is a PPT algorithm that takes a matrix $\boldsymbol{B} \in \mathbb{Z}_q^{n\times m'}$, whose first m columns span $\mathbb{Z}_q^n$, and a basis $\boldsymbol{T_A}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$, where $\boldsymbol{A}$ is the left $n \times m$ submatrix of $\boldsymbol{B}$ as inputs, and outputs a basis $\boldsymbol{T_B}$ of $\Lambda_q^{\perp}(\boldsymbol{B})$ with $||\widetilde{\boldsymbol{T_B}}|| \leq ||\widetilde{\boldsymbol{T_A}}||$.*

Lemma 4 ([1, Theorem. 3]). *SampleRight is a PPT algorithm that takes matrices* $\boldsymbol{A}, \boldsymbol{C} \in \mathbb{Z}_q^{n\times m}$*, a low-norm matrix* $\boldsymbol{R} \in \mathbb{Z}^{m\times m}$*, a short basis* $\boldsymbol{T_C} \in \mathbb{Z}^{m\times m}$ *of* $\Lambda_q^{\perp}(\boldsymbol{C})$*, a vector* $\boldsymbol{u} \in \mathbb{Z}_q^n$ *and a rational* s *such that* $s \geq ||\widetilde{\boldsymbol{T_C}}|| \cdot \Omega(\sqrt{\log n})$ *as inputs, and outputs vectors* $\boldsymbol{b} \in \mathbb{Z}^{2m}$*, such that* $[\boldsymbol{A} \mid \boldsymbol{A}\cdot\boldsymbol{R}+\boldsymbol{C}]\cdot\boldsymbol{b} = \boldsymbol{u} \mod q$ *and with distribution statistically close to* $D_{\Lambda,s}$*, where* Λ *denotes the shifted lattice* $\{\boldsymbol{x} \in \mathbb{Z}^{2m} : [\boldsymbol{A} \mid \boldsymbol{A}\cdot\boldsymbol{R}+\boldsymbol{C}]\cdot\boldsymbol{x} = \boldsymbol{u} \mod q\}$.

3 Coping with VLR for Libert's Dynamic Group Signature Scheme from Lattices

This section first recalls the scheme given in [20] in brief, which used the syntax and security model of Kiayias and Yung [18]. Then this section discusses the complications of incorporating VLR with the group signature schemes based on lattices and how to achieve the problems with a justified scheme.

The "power-of-2" matrix $\mathbf{H}_{n\times n\lceil\log q\rceil} \in \mathbb{Z}_q^{n\times n\lceil\log q\rceil}$, for any positive integers n, and $q \geq 2$ is given as
$\mathbf{H}_{n\times n\lceil\log q\rceil} = \mathbf{I}_n \otimes [1\,|\,2\,|\,4\,|...|\,2^{\lceil\log q\rceil-1}] =$

$$\begin{bmatrix} 1\,2\,4\,\ldots\,2^{\lceil\log q\rceil-1} & & & \\ & 1\,2\,4\,\ldots\,2^{\lceil\log q\rceil-1} & & \\ & & \ddots & \\ & & & 1\,2\,4\,\ldots\,2^{\lceil\log q\rceil-1} \end{bmatrix}.$$

Moreover, for each vector $\mathbf{v} = \mathbf{H}_{n\times n\lceil\log q\rceil} \cdot \mathsf{bin}(\mathbf{v}) \in \mathbb{Z}_q^n$, where $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{n\lceil\log q\rceil}$ is the binary expression of $\mathbf{v}$ and $\mathsf{bin}(\mathbf{v})$ is obtained by replacing each coordinate of $\mathbf{v}$.

The key component of the scheme given in [20] is the two-message joining protocol. Through the joining-protocol, new users can join the group and the group manager can grant the member certifications. First new user $User_i$, who is having a long-term public and private key pair ($\mathsf{upk}[i]$ and $\mathsf{usk}[i]$) samples a secret signing key $\mathbf{x}_i \hookleftarrow D_{\mathbb{Z}^{4m},\sigma}$, which is a short vector and used to compute a syndrome $\mathbf{v}_i = \mathbf{F}\cdot\mathbf{x}_i \in \mathbb{Z}_q^{4n}$ (where $m = 2n\lceil\log q\rceil$ and $\mathbf{F} \in \mathbb{Z}_q^{4n\times 4m}$). The group manager signs $\mathsf{bin}(\mathbf{v}_i)$ the binary expression of $\mathbf{v}_i$ to generate the certification. Finally, the group manager sends the triple $(id_i, \mathbf{d}_i, \mathbf{s}_i)$ to the new user $User_i$, where id_i is the ℓ-bit identifier selected for the new user and $\mathbf{d}_i$ is the computed short vector using the sampled short vector $\mathbf{s}_i$. The user $User_i$ can sign a message with his secret signing key $\mathbf{x}_i$ and his member certificate $(id_i, \mathbf{d}_i, \mathbf{s}_i)$. By using a Stern-like protocol, $User_i$ can prove he has a valid certificate, which is associated with the public key $\mathbf{v}_i$.

However, when applying Verifier-local revocation method to a scheme and trying to achieve full-security, two main points should be take care. (1) The revocation tokens (especially the challenged indices' tokens) should not be given to the adversary since he can execute Verify with those tokens and identify the signer of the challenging signature. (2) The revocation tokens should not be a part of the secret signing key. Since at the full-anonymity defined in the BMW03 model, the almost-full anonymity defined in [25], and the dynamical-almost-full anonymity given in [24], all the secret signing keys are given to the adversary, and the adversary can extract the revocation tokens from the secret signing keys easily if we generate revocation token as a part of the secret signing key. When we apply VLR to the scheme in [20], we can use the almost-full anonymity given in [25] as a solution for the case (1). The almost-full anonymity provides all the secret signing keys and the group public key to the adversary at the beginning of the game as in the full-anonymity game [4]. But, the revocation tokens are given only upon the request of the adversary. Moreover, revocation tokens are not provided, which are used for generating the challenging signature, and the challenging signature is not generated for the indices, whose revocation tokens are revealed. Then we use a vector $\mathbf{d}_i$ to make the revocation token of the new scheme. Since $\mathbf{d}_i$ should satisfy some computation with $\mathsf{bin}(\mathbf{v}_i)$ and some other parameters, it has a connection to the identifier and the public key of the signer. Hence, the signers cannot forge $\mathbf{d}_i$. Accordingly, $\mathbf{d}_i$ is suitable for the revocation token. Moreover, for member revocation, the group manager should know the revocation token of the revoking member. Since $\mathbf{d}_i$ is generated by the group manager he can create the revocation token and provide with the member certificate. Using $\mathbf{d}_i$ for creating revocation token is suitable and it is the solution for the concern (2).

Moreover, when dealing with fully dynamic group signature scheme with member registration, we should allow the adversary to join the group as a new member. At the joining protocol, the group manager provides the certification including the revocation token to the new users. By using this information, the adversary can attack later at the challenging phase. The dynamical-almost-full anonymity suggested in [24] which is an extended version of the almost-full anonymity for fully-dynamic group signature schemes provides a solution for this matter. In the dynamical-almost-full anonymity, when the adversary joins the group as a new user, the revocation token will not be provided. However, the adversary can request any revocation token (including newly added ones) except revocation tokens of the indices used to generate challenging signature (as in the almost-full anonymity). Moreover, at the challenging phase, the adversary can only use the indices which are added by him at the registration query. Thus, the adversary cannot cheat using the user details added before the game as a legal member.

The dynamical-almost-full anonymity game between a challenger C and an adversary A 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:** A can join the group as a new user any number of time via registration query. C generates revocation token and certificate for the new user if the new user is valid. Then C saves the 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, member certification *cert* is sent without the revocation token. Moreover, A can query revocation token ($\mathbf{grt}$) of any user and can access the opening oracle with any message M and a valid signature Σ.

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

Our scheme uses the dynamical-almost-full anonymity to ensure the security.

4 New scheme

The construction of the new scheme is same as the dynamic lattice-based group signature scheme suggested in [20], but with member revocation facility using VLR. Thus, the group manager can revoke misbehaved members other than providing member certifications. In such a way, our scheme offers both member registration and member revocation. However, we present our scheme by highlighting the differences between our scheme and the scheme given in [20].

Our scheme consists of six algorithms; Setup, Join, Sign, Verify, Open, and Revoke. In the beginning, the group public key and the authority keys are generated in Setup. A new user, who wants to join the group should interact with the group manager using Join. If the key provided by the new user is valid, then the group manager issues the member certification. In the scheme given in [20], the member certification is $(id_i, \mathbf{d}_i, \mathbf{s}_i)$, where id_i is the identifier of the new member, and $\mathbf{d}_i$ and $\mathbf{s}_i$ are short vectors. Here we use the short vector $\mathbf{d}_i$ to generate the new member's revocation token $\mathbf{grt}[i] = \mathbf{A} \cdot \mathbf{r}_i$, where $\mathbf{r}_i$ is an element of $\mathbf{d}_i$. Thus, member certification issued by the group manager will be $(id_i, \mathbf{d}_i, \mathbf{s}_i, \mathbf{grt}[i])$ in our scheme. When a member wants to generate a signature, he has to compute $\mathbf{v} = \mathbf{V} \cdot (\mathbf{A} \cdot \mathbf{r}_i) + \mathbf{e}_1 \mod q$ other than the computations given in [20]. At the verification stage of the signature (in Verify), the verifiers check the validity of the signer by screening the revocation list he has. Authenticating the signer is not given in [20] because they have not considered the member revocation. We use the algorithms Setup and Open given in [20] without any change. However, we provide a new algorithm called Revoke to cancel the membership of the misbehaved members.

4.1 Description of Our Scheme

Setup: The randomized algorithm $\mathsf{KeyGen}(1^n, 1^m)$ works as follows.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T_A}$. Then sample random matrices $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n\times m}), \mathbf{D}_0, \mathbf{D}_1 \hookleftarrow U(\mathbb{Z}_q^{2n\times 2m})$ and a vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.
2. Select an additional random matrix $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{4n\times 4m})$.
3. Generate a master key pair; a statistically uniform matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T_B} \in \mathbb{Z}^{m\times m}$ for the GPV-IBE [15] scheme in its multi-bit variant. The basis $\mathbf{T_B}$ allows to compute GPV private keys with a Gaussian parameter $\sigma_{GPV} \geq ||\widetilde{\mathbf{T_B}}|| \cdot \sqrt{\log m}$.
4. Choose a one-time signature scheme $\mathcal{OTS} = (\mathsf{OGen}, \mathsf{OSign}, \mathsf{OVer})$, and a hash function $\mathcal{H}_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q^{n\times 2m}$.
5. Finally, we have
   the group public key $\mathbf{gpk}{:=}\ (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{B}, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{F}, \mathbf{u}, \mathcal{OTS}, \mathcal{H}, \mathcal{H}_0)$, the group manager's (issuer's) secret key $\mathbf{ik}{:=}\ \mathbf{T_A}$ and the opener's secret key $\mathbf{ok}{:=}\ \mathbf{T_B}$.

Join: A new user $User_i$, who has a personal public and private key pair $(\mathbf{upk}[i], \mathbf{usk}[i] \leftarrow \mathsf{UKg}(1^n))$ interacts with the group manager GM to join the group through the joining protocol.

1. $User_i$ samples a discrete Gaussian vector $\mathbf{x}_i \leftarrow D_{\mathbb{Z}^{4m},\sigma}$, and computes $\mathbf{z}_i \leftarrow \mathbf{F} \cdot \mathbf{x}_i \in \mathbb{Z}_q^{4n}$, where $\mathbf{x}_i$ is the secret signing key $(\mathbf{gsk}[i])$ of $User_i$. Then he generates an ordinary digital signature $\Sigma_{join} \leftarrow \mathsf{Sig}(\mathbf{usk}[i], \mathbf{z}_i)$, whose binary representation $\mathsf{bin}(\mathbf{z}_i)$ consists of $4n\lceil\log q\rceil = 2m$ bits, and sends $\mathbf{z}_i$ and Σ_{join} to the group manager GM.
2. The group manager verifies that $\mathbf{z}_i$ was not previously used by any user using the registration table *reg* and he verifies Σ_{join} is a valid signature on $\mathbf{z}_i$, using $\mathsf{Vf}(\mathbf{upk}[i], \mathbf{z}_i, \Sigma_{join})$. He aborts if any condition fails. Otherwise, the group manager selects a fresh ℓ-bit string $id_i = id_i[1] \ldots id_i[\ell] \in \{0,1\}^\ell$ as the index of the user $User_i$. Then GM certifies the new user $User_i$ as a new member and generates the member certification as below.
   First, GM defines a matrix for $User_i$,

$$\mathbf{A}_{id_i} = \left[\mathbf{A}|\mathbf{A}_0 + \sum_{j=1}^{\ell} id_i[j]\mathbf{A}_j\right] \in \mathbb{Z}_q^{n\times 2m}. \tag{1}$$

   Next, GM executes $\mathsf{ExtBasis}(\mathbf{A}_{id_i}, \mathbf{T_A})$ to obtain a short delegated basis $\mathbf{T}'_{id_i}$ of $\Lambda_q^{\perp}(\mathbf{A}_{id_i}) \in \mathbb{Z}^{2m\times 2m}$.
   Then, GM choses a short vector $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$, and uses delegated basis $\mathbf{T}'_{id_i}$ to compute short vector $\mathbf{d}_i = \left[\frac{\mathbf{d}_{i,1}}{\mathbf{d}_{i,2}}\right] \in \mathbb{Z}^{2m}$ such that

$$\begin{aligned} \mathbf{A}_{id_i}\mathbf{d}_i &= \left[\mathbf{A}|\mathbf{A}_0 + \sum_{j=1}^{\ell} id_i[j]\mathbf{A}_j\right] \cdot \mathbf{d}_i \\ &= \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{z}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i) \mod q. \end{aligned} \tag{2}$$

   After that, GM selects $\mathbf{d}_{i,1}$ or $\mathbf{d}_{i,2}$ randomly as $\mathbf{r}_i$ and generates the revocation token $\mathbf{grt}[i]{=}(\mathbf{A}\cdot\mathbf{r}_i)$ and member certification $cert_i = (id_i, \mathbf{d}_i, \mathbf{s}_i, \mathbf{grt}[i])$.

Finally, GM saves the new member's details $(\mathbf{z}_i, cert_i, i, \mathbf{upk}[i], \Sigma_{join})$ and sends the certification $cert_i = (id_i, \mathbf{d}_i, \mathbf{s}_i, \mathbf{grt}[i])$ to the new user.

Sign: $\mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], cert_i, M)$ is a randomized algorithm, that generates a signature Σ on a given message M using $\mathbf{gsk} = \mathbf{x}_i \in \mathbb{Z}^{4m}$ and $cert_i$ as follows.

1. Parse $cert_i$ as $(id_i, \mathbf{d}_i, \mathbf{s}_i, \mathbf{grt}[i])$, where $\mathbf{d}_i = [\mathbf{d}_{i,1}^T | \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}_q^{2m}$, $\mathbf{s}_i \in \mathbb{Z}^{2m}$ and $\mathbf{grt}[i] = (\mathbf{A} \cdot \mathbf{r}_i)$.
2. Generate one-time signature key pair as $\mathsf{OGen}(1^n) \rightarrow (\mathbf{ovk}, \mathbf{osk})$.
3. Encrypt the index $d = \mathsf{bin}(\mathbf{z}_i)$, where $\mathbf{z}_i = \mathbf{F} \cdot \mathbf{x}_i$ and compute $\mathbf{c}_{\mathbf{z}_i} \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}$.
   (a) Let $\mathbf{G} = \mathcal{H}_0(\mathbf{ovk}) \in \mathbb{Z}_q^{n \times 2m}$.
   (b) Sample $\mathbf{e}_0 \leftarrow \chi^n$, $\mathbf{e}_1 \leftarrow \chi^m$ and $\mathbf{e}_2 \leftarrow \chi^{2m}$.
   (c) Compute the ciphertext $\mathbf{c}_{\mathbf{z}_i}$
   $$\mathbf{c}_{\mathbf{z}_i} = (\mathbf{c}_1, \mathbf{c}_2) = (\mathbf{B}^T \mathbf{e}_0 + \mathbf{e}_1, \mathbf{G}^T \mathbf{e}_0 + \mathbf{e}_2 + \mathsf{bin}(\mathbf{z}_i) \lfloor q/2 \rfloor). \qquad (3)$$
4. Sample $\rho \xleftarrow{\$} \{0,1\}^n$, let $\mathbf{V} = \mathcal{G}(\mathbf{A}, \mathbf{u}, M, \rho) \in \mathbb{Z}_q^{m \times n}$ and compute $\mathbf{v} = \mathbf{V} \cdot (\mathbf{A} \cdot \mathbf{r}_i) + \mathbf{e}_1 \mod q$ ($\mathcal{G} : \{0,1\}^* \rightarrow \mathbb{Z}_q^{n \times m}$ is a random oracle and $||\mathbf{e}_1||_\infty \leq \beta$ with overwhelming probability).
5. Use the protocol given in Section 4.2 to prove the knowledge of $id_i \in \{0,1\}^\ell$, vectors $\mathbf{s}_i \in \mathbb{Z}^{2m}, \mathbf{d}_{i,1}, \mathbf{d}_{i,2} \in \mathbb{Z}^m, \mathbf{x}_i \in \mathbb{Z}^{4m}$ with infinity norm bound β; $\mathbf{e}_0 \in \chi^n, \mathbf{e}_1 \in \chi^m, \mathbf{e}_2 \in \chi^{2m}$ with infinity norm bound b and $\mathsf{bin}(\mathbf{z}_i) \in \{0,1\}^{2m}, \mathbf{w}_i \in \{0,1\}^m$, that satisfy equation (3) and
   $\mathbf{A} \cdot \mathbf{d}_{i,1} + \mathbf{A}_0 \cdot \mathbf{d}_{i,2} + \sum_{j=1}^{\ell} (id_i[j] \cdot \mathbf{d}_{i,2}) \cdot \mathbf{A}_j - \mathbf{D} \cdot \mathbf{w}_i = \mathbf{u} \in \mathbb{Z}_q^n$ and
   $$\begin{cases} \mathbf{H}_{2n \times m} \cdot \mathbf{w}_i = \mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{z}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i \in \mathbb{Z}_q^{2n} \\ \mathbf{F} \cdot \mathbf{x}_i = \mathbf{H}_{4n \times 2m} \cdot \mathsf{bin}(\mathbf{z}_i) \in \mathbb{Z}_q^{4n} \\ \mathbf{V} \cdot (\mathbf{A} \cdot \mathbf{r}_i) + \mathbf{e}_1 = \mathbf{v} \mod q. \end{cases}$$
   Repeat the protocol $t = \omega(\log n)$ times to make the soundness error negligible. 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 $= (\{Ch^{(k)}\}_{k=1}^t) = \mathcal{H}(M, \mathbf{ovk}, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_{\mathbf{z}_i})$.
6. Compute one-time signature $sig = \mathsf{OSig}(\mathbf{osk}, (\mathbf{c}_{\mathbf{z}_i}, \Pi))$.
7. Output signature $\Sigma = (\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$.

Verify: The deterministic algorithm $\mathsf{Verify}(\mathbf{gpk}, M, \Sigma, RL)$, where $RL = \{\{\mathbf{u}_i\}_i\}$ works as follows.

1. Parse the signature Σ as $(\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$.
2. Get $\mathbf{V} = \mathcal{G}(\mathbf{A}, \mathbf{u}, M, \rho) \in \mathbb{Z}_q^{m \times n}$.
3. If $\mathsf{OVer}(\mathbf{ovk}, (\mathbf{c}_{\mathbf{z}_i}, \Pi), sig) = 0$ then return 0.
4. Parse Π as $(\{CMT^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{RSP^{(k)}\}_{k=1}^t)$.
5. If $(Ch^{(1)}, ..., Ch^{(t)}) \neq \mathcal{H}(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$ return 0 else proceed.
6. 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.

7. 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.
8. Return valid.

Open: $\mathsf{Open}(\mathbf{gpk}, \mathbf{ok}, M, \Sigma, reg)$ functions as below.

1. Parse $\mathbf{ok} = \mathbf{T_B}$ and Σ as $(\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$.
2. Let $\mathbf{G} = \mathcal{H}_0(\mathbf{ovk}) \in \mathbb{Z}_q^{n \times 2m}$.
3. Using $\mathbf{T_B}$ compute a small norm matrix $\mathbf{Y} \in \mathbb{Z}^{m \times 2m}$, where $\mathbf{B} \cdot \mathbf{Y} = \mathbf{G} \mod q$.
4. Compute $\mathbf{bin}(\mathbf{z}) = \lfloor (\mathbf{c}_2 - \mathbf{Y}^T \cdot \mathbf{c}_1)/(q/2) \rceil$.
5. Determine whether the obtained $\mathbf{bin}(\mathbf{z})$ is corresponding to a vector $\mathbf{z} = \mathbf{H}_{4n \times 2m} \cdot \mathbf{bin}(\mathbf{z}) \mod q$ in reg and output the corresponding index i.

Revoke: The algorithm $\mathsf{Revoke}(\mathbf{gpk}, \mathbf{ik}, i, reg)$ functions as follows.

1. Query reg for i and obtain revoking member's revocation token $(\mathbf{A} \cdot \mathbf{r}_i)$.
2. Add $(\mathbf{A} \cdot \mathbf{r}_i)$ to RL and update the registration table $reg[i]$ to inactive (0).
3. Return RL and reg.

4.2 The Underlying ZKAoK for the Group Signature Scheme

Let COM be the statistically hiding and computationally binding commitment scheme described in [17]. The common inputs are matrices $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{D}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}, \mathbf{D}_0, \mathbf{D}_1 \in \mathbb{Z}_q^{2n \times 2m}, \mathbf{F} \in \mathbb{Z}_q^{4n \times 4m}, \mathbf{H}_{2n \times m} \in \mathbb{Z}_q^{2n \times m}, \mathbf{H}_{4n \times 2m} \in \mathbb{Z}_q^{4n \times 2m}, \mathbf{G} \in \mathbb{Z}^{n \times 2m}, \mathbf{V} \in \mathbb{Z}^{m \times n}$ and vectors $\mathbf{u} \in \mathbb{Z}_q^n, \mathbf{c}_1 \in \mathbb{Z}_q^m, \mathbf{c}_2 \in \mathbb{Z}_q^{2m}, \mathbf{v} \in \mathbb{Z}_q^n$. The prover's inputs are $\mathbf{x} \in [-\beta, \beta]^{4m}, \mathbf{y} \in \{0,1\}^{2m}, \mathbf{w} \in \{0,1\}^m, \mathbf{d}_1, \mathbf{d}_2 \in [-\beta, \beta]^m, \mathbf{s} \in [-\beta, \beta]^{2m}, id = (id[1], \ldots, id[\ell])^T \in \{0,1\}^\ell, \mathbf{e}_0 \in [-b, b]^n, \mathbf{e}_1 \in [-b, b]^m, \mathbf{e}_2 \in [-b, b]^{2m}, \mathbf{r} \in [-\beta, \beta]^m$. The prover's goal is to convince the verifier in ZK that

$$\begin{cases} \mathbf{F} \cdot \mathbf{x} = \mathbf{H}_{4n \times 2m} \cdot \mathbf{y} \mod q; \mathbf{H}_{2n \times m} \cdot \mathbf{w} = \mathbf{D}_0 \cdot \mathbf{y} + \mathbf{D}_1 \cdot \mathbf{s} \mod q; \\ \mathbf{A} \cdot \mathbf{d}_1 + \mathbf{A}_0 \cdot \mathbf{d}_2 + \sum_{j=1}^{\ell} \mathbf{A}_j \cdot (id[j] \cdot \mathbf{d}_2) - \mathbf{D} \cdot \mathbf{w} = \mathbf{u} \mod q; \\ \mathbf{c}_1 = \mathbf{B}^T \mathbf{e}_0 + \mathbf{e}_1 \mod q; \mathbf{c}_2 = \mathbf{G}^T \mathbf{e}_0 + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \mathbf{y} \mod q; \\ \mathbf{V} \cdot (\mathbf{A} \cdot \mathbf{r}) + \mathbf{e}_1 = \mathbf{v} \mod q. \end{cases}$$

We use the interacting protocol given in [20]. To prove $\mathbf{V} \cdot (\mathbf{A} \cdot \mathbf{r}) + \mathbf{e}_1 = \mathbf{v} \mod q$ we use the proof given in [26].

5 Correctness and Security Analysis of the Scheme

5.1 Correctness

1. Assume both the group manager and the new user follow the joining protocol honestly and communicate via a secured channel. The group manager verifies whether the public key of the new user is not being used before, and issues the member-certificate with revocation token only for valid users.

2. For all $\mathbf{gpk}$, $\mathbf{gsk}$, and $\mathbf{grt}$,
   $\mathsf{Verify}(\mathbf{gpk}, M, \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], (id_i, \mathbf{d}_i, \mathbf{s}_i, \mathbf{grt}[i]), M), RL) =$ Valid and $\mathbf{grt}[i] \notin RL$.
   $\mathsf{Open}(\mathbf{gpk}, \mathbf{ok}, M, \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], (id_i, \mathbf{d}_i, \mathbf{s}_i, \mathbf{grt}[i]), M), reg) = i.$

Verify in the proposed scheme only accepts signatures generated on given messages and which are only generated by active (not revoked) and honest users (has member certificate). If the revocation token of the signer is in RL, then his signature is not accepted by Verify. Similarly Sign also checks whether the signer can satisfy those requirements. The signer has to convince the verifier his validity using the zero-knowledge protocol. Zero-knowledge protocol guarantees no one can sign and pass the verification of signing process without having a valid membership and secret signing key. The algorithm Open outputs the index of the signer with overwhelming probability. It computes $\mathsf{bin}(\mathbf{z}_i)$ and verifies with the registration table *reg*.

5.2 Anonymity

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

Here a sequence of games between the challenger and the adversary is used, where the advantage of the adversary is negligible in the last game.

Game 0: This is the real experiment. The challenger C runs $\mathsf{KeyGen}(1^n, 1^N)$ to obtain the group public key and the authorities' keys. The challenger C gives the group public key $\mathbf{gpk}$ and all the existing group members' secret keys $\mathbf{gsk}$ to the adversary A. However any revocation token information is not given to A at the beginning. In the query phase, A can join as a new member any number of time through the registration query. In the registration query, C will accept valid members but will provide the certification $cert = (id_i, \varepsilon, \mathbf{s}_i, \varepsilon)$ without new user's revocation token or related details. However, A can request for revocation tokens of any member, and he can access opening query for any signature. In the challenge phase, A sends two indices (i_0, i_1) together with a message M^*. If (i_0, i_1) are newly added by A and if (i_0, i_1) are not used for querying revocation tokens, then C generates and sends back the challenging signature $\Sigma^* = (\mathbf{ovk}^*, \mathbf{c}^*_{\mathbf{z}_i}, \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ for a random bit $b \leftarrow \{0,1\}$. The adversary's goal is to identify which index is used to generate the challenging signature. A returns b'. If $b' = b$ then the experiment returns 1. Otherwise, returns 0.

Game 1: In this game, the challenger C makes a slight modification with respect to Game 0. In real experiment (Game 0) one-time key pair ($\mathbf{ovk}$, $\mathbf{osk}$) is generated at the signature generation. In this game, C generates the one-time key pair $(\mathbf{ovk}^*, \mathbf{osk}^*)$ at the beginning of the game. If the adversary A accesses the opening oracle with a valid signature $\Sigma = (\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$, where $\mathbf{ovk}$=$\mathbf{ovk}^*$, C returns a random bit and aborts. However, A comes up with a signature Σ, where $\mathbf{ovk}$=$\mathbf{ovk}^*$ contradicts the strong unforgeability of $\mathcal{OTS}$, and since $\mathbf{ovk}^*$ is independent of the adversary's view, probability of

$\mathbf{ovk}$=$\mathbf{ovk}^*$ is negligible. Even after seeing the challenging signature if A comes up with a valid signature $\Sigma = (\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$, where $\mathbf{ovk}$=$\mathbf{ovk}^*$, then sig is a forged one-time signature, which defeats the strong unforgeability of $\mathcal{OTS}$. Thus, we assume that A does not request for opening of a valid signature with $\mathbf{ovk}^*$ and the challenger aborting the game is negligible.

Game 2: The challenger C programs the random oracle $\mathcal{H}_0$. At the beginning of the game, C replaces $\mathbf{G}$. C chooses uniformly random matrix $\mathbf{G}^* \in \mathbb{Z}_q^{n\times 2m}$ and sets $\mathcal{H}_0(\mathbf{ovk}^*)$=$\mathbf{G}^*$. To answer the opening oracle requests of A with $\Sigma = (\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$, C samples a small-norm matrix $\mathbf{Y} \leftarrow D^{2m}_{z^m,\sigma}$, and computes $\mathbf{G} = \mathbf{B} \cdot \mathbf{Y} \mod q$. 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 $\mathcal{H}_0(\mathbf{ovk})$. Since for the view of A, the distribution of $\mathbf{G}^*$ is statistically close to the real experiment [15], Game 2 is indistinguishable from Game 1.

Game 3: Instead of honestly generating the legitimate non-interactive proof Π, the challenger C simulates the proof without using the witness. C invokes the simulator for each $k \in [t]$ and then programs the random oracle $\mathcal{H}$ accordingly. The challenging signature $\Sigma^* = (\mathbf{ovk}^*, \mathbf{c}^*_{\mathbf{z}_i}, \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ is statistically close to the challenging signature in Game 2 because the argument system is statistically zero-knowledge. Thus Game 3 is indistinguishable from Game 2.

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

Game 5: Game 4 has $\mathbf{v} = \mathbf{V} \cdot \mathbf{t} + \mathbf{e}_1 \mod q$. In this game the challenger C makes $\mathbf{v}$ truly uniform by sampling $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ and setting $\mathbf{v} = \mathbf{y}$. Thus, C makes revocation token totally independent of the bit b. In Game 4, ($\mathbf{V}$, $\mathbf{v}$) pair is a proper $LWE_{n,q,\chi}$ instance. Thus, the distribution of the pair ($\mathbf{V}$, $\mathbf{v}$) is computationally close to the uniform distribution over $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$. Game 4 and Game 5 are indistinguishable under the assumption of the hardness of $LWE_{n,q,\chi}$ problem. If the adversary can distinguish $\mathbf{v}$ and $\mathbf{y}$, then he can solve Decision-LWE problem.

Game 6: In this game the challenger C modifies the generation of ciphertext $\mathbf{c}_{z_i} = (\mathbf{c}_1^*, \mathbf{c}_2^*)$ 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 and 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 Decision-LWE problem. That means, he can distinguish $(\mathbf{B}^*, (\mathbf{B}^*)^T\mathbf{e}_0 + \mathbf{e}_1)$ from $(\mathbf{B}^*, \mathbf{z}_1)$ and $(\mathbf{G}^*, (\mathbf{G}^*)^T\mathbf{e}_0 + \mathbf{e}_2)$ from $(\mathbf{G}^*, \mathbf{z}_2)$ which conflicts with $LWE_{n,q,\chi}$ assumption.

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

Hence, these games prove that our scheme is secure with dynamical-almost-full anonymity, which applied for fully dynamicity.

5.3 Traceability

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

Let B be a PPT algorithm that solves SIS problem with non-negligible probability. The adversary A, who has **gpk** and **ok** outputs (M, Σ) in the traceability game. He can add new users and replace members' personal public keys. Moreover, he can query for secret signing keys and revocation tokens of any member. For the queries of A, B answers as in [21] and [20]. In [20], first B selects $coins \hookleftarrow U(\{0,1,2\})$ as a guess for the misidentification attacks that A will mount. The case $coin = 0$ corresponds, when the knowledge extractor of the proof system reveals witnesses after repeated executions of A and witnesses containing a new identifier $id^* \in \{0,1\}^\ell$ that does not belong to any user. The case $coin = 1$ corresponds to when B expects that the knowledge extractor will obtain the identifier $id^* = id^\dagger$ of a group member in the group. The case $coin = 2$ corresponds to when B is expecting decrypting $\mathbf{c}^*_{\mathbf{z}_i}$ and knowledge extractor will disclose vectors $\mathsf{bin}(\mathbf{z}^*), \mathbf{w}$, and $\mathbf{s}$. Depending on $coin \in \{0,1,2\}$, the group public key is generated using different methods and methods of answering to the queries of A also different as per $coin$.

Finally, A outputs a forgery signature $\Sigma^*{=}(\mathbf{ovk}^*, \mathbf{c}^*_{\mathbf{z}_i}, \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ on message M^*. B opens Σ^* and obtains the index. As same as in [21] and [20], the improved Forking Lemma [11] 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 [20] and finally we can say, if A has non-negligible success probability and runs in polynomial time, then so does B. This concludes our proof of traceability.

5.4 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, active, and honest user i (i 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 and authority keys. 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 $\Sigma{=}(\mathbf{ovk}, \mathbf{c}_{\mathbf{z}_i}, \Pi, sig, \mathbf{v}, \rho)$.

Finally, A outputs $\mathit{\Sigma}^*{=}(\mathbf{ovk}^*, \mathbf{c}^*_{\mathbf{z}_i}, \mathit{\Pi}^*, sig^*, \mathbf{v}^*, \rho^*)$ signed on a message M^* and which opens to i^* who did not sign the message. Thus, $(M^*, \mathit{\Sigma}^*)$ 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 $\mathbf{z}_{i\prime} = \mathbf{F} {\cdot} \mathbf{x}_{i\prime} \mod q$. To compute such a vector, B proceeds by replaying A sufficient times and applying Improved Forking Lemma [11]. As discussed in [20], from the corresponding responses of $\mathit{\Pi}^*$, B can extract a short vector $\mathbf{x}\prime$, where $\mathbf{z}_{i^*} = \mathbf{F} \cdot \mathbf{x}\prime \mod q$. According to the Stern-like proof of knowledge, with overwhelming probability, we say $\mathbf{x}\prime \neq \mathbf{x}_{i^*}$.

This proves the non-frameability of proposed scheme.

6 Conclusion

This paper showed how to obtain member revocation with VLR to the existing member registration scheme [20]. We provided a revocation token generation method that uses a current attribute of the existing scheme. Moreover, we proved the security of the new scheme with the dynamical-almost-full anonymity. For the underlying interactive protocol, we used the protocol given in [20] with the proof of the signer's revocation token which is committed via an LWE function.

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. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (h) ibe in the standard model. In: EUROCRYPT 2010. vol. 6110. Springer (2010)
2. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. STACS 2009 pp. 75–86 (2009)
3. Ateniese, G., Song, D., Tsudik, G.: Quasi-efficient revocation of group signatures. In: Proceedings of Financial Cryptography 2002, LNCS. pp. 183–197. Springer (2002)
4. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: EUROCRYPT 2003, LNCS. vol. 2656, pp. 614–629. Springer Berlin Heidelberg (2003)
5. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: The case of dynamic groups. In: CT-RSA 2005. vol. 3376, pp. 136–153. LNCS (2005)
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO 2004, LNCS. pp. 41–55. Springer (2004)
7. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: ACM-CCS 2004. pp. 168–177. ACM (2004)
8. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: ACNS 2016, LNCS. vol. 9696, pp. 117–136. Springer, Cham (2016)
9. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC 2013. pp. 575–584. ACM (2013)

10. Brickell, E.: An efficient protocol for anonymously providing assurance of the container of the private key. *Submitted to the Trusted Comp. Group (April 2003)* (2003)
11. Brickell, E., Pointcheval, D., Vaudenay, S., Yung, M.: Design validations for discrete logarithm based signature schemes. In: PKC 2000, LNCS. vol. 1751, pp. 276–292. Springer Berlin Heidelberg (2000)
12. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: CRYPTO 2002, LNCS. vol. 2442, pp. 61–76. Springer Berlin Heidelberg (2002)
13. Camenisch, J., Neven, G., Rückert, M.: Fully anonymous attribute tokens from lattices. In: SCN 2012, LNCS. vol. 12, pp. 57–75. Springer Berlin Heidelberg (2012)
14. Chaum, D., Van Heyst, E.: Group signatures. In: EUROCRYPT 1991, LNCS. vol. 547, pp. 257–265. Springer Berlin Heidelberg (1991)
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: ACM 2008. pp. 197–206. ACM (2008)
16. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: ASIACRYPT 2010, LNCS. vol. 6477, pp. 395–412. Springer Berlin Heidelberg (2010)
17. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: ASIACRYPT 2008, LNCS. vol. 5350, pp. 372–389. Springer Berlin Heidelberg (2008)
18. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. International Journal of Security and Networks 1(1-2), 24–45 (2006)
19. Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-based group signature scheme with verifier-local revocation. In: PKC 2014, LNCS. vol. 8383, pp. 345–361. Springer Berlin Heidelberg (2014)
20. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In: ASIACRYPT 2016, LNCS. vol. 10032, pp. 373–403. Springer Berlin Heidelberg (2016)
21. Ling, S., Nguyen, K., Wang, H.: Group signatures from lattices: simpler, tighter, shorter, ring-based. In: PKC 2015, LNCS. vol. 9020, pp. 427–449. Springer Berlin Heidelberg (2015)
22. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Lattice-based group signatures: Achieving full dynamicity with ease. In: ACNS 2017, LNCS. vol. 10355, pp. 293–312. Springer International Publishing, Cham (2017)
23. Peikert, C.: A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science 10(4), 283–424 (2016), https://doi.org/10.1561/0400000074
24. Perera, M.N.S., Koshiba, T.: Achieving almost-full security for lattice-based fully dynamic group signatures with verifier-local revocation. In: ISPEC 2018, LNCS (to appear)
25. Perera, M.N.S., Koshiba, T.: Fully dynamic group signature scheme with member registration and verifier-local revocation. In: ICMC 2018, Mathematics and Computing (to appear)
26. Perera, M.N.S., Koshiba, T.: Zero-knowledge proof for lattice-based group signature schemes with verifier-local revocation. In: 9-th International Workshop on Trustworthy Computing and Security (TwCSec-2018), LNDT (to appear)
27. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005. pp. 84–93. ACM Press (2005)