*cryptography*

MDPI

*Article*

# Almost Fully Secured Lattice-Based Group Signatures with Verifier-Local Revocation †

**Maharage Nisansala Sevwandi Perera [1,*] and Takeshi Koshiba [2]**

[1] Adaptive Communications Research Laboratories, Advanced Telecommunications Research Institute International (ATR), Kyoto 619-0288, Japan

[2] Faculty of Education and Integrated Arts and Sciences, Waseda University, Tokyo 169-8050, Japan; tkoshiba@waseda.jp

* Correspondence: perera.nisansala@atr.jp or mnisansalasperera@gmail.com

† The paper is an extended version of our paper published in AINA2017. The preliminary version of this work was appeared in the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA). This paper extends the contribution in AINA2017 with some fixes which were not captured in AINA2017 paper.

Received: 29 October 2020; Accepted: 25 November 2020; Published: 30 November 2020


**Abstract:** An efficient member revocation mechanism is a desirable feature when group signature schemes are applied in practical scenarios. Revocation methods, such as verifier-local revocation (VLR), provide an efficient member revocation in applications of group signatures. However, VLR-group signatures rely on a weaker security notion. On the other hand, group signature schemes for static groups gain stronger security with the full-anonymity security notion. Even though an outsider sees the secret signing keys of all group members in the full-anonymity, the signer is still anonymous. Achieving the full-anonymity for VLR group signature schemes is challenging due to the structure of secret signing keys. The secret signing keys of those schemes consist of tokens, which are used to manage revocation. The reveal of tokens may destroy the anonymity of the signers. We obtain stronger security for the lattice-based VLR group signature schemes by providing a new key generation method, which outputs revocation tokens without deriving from the members' secret signing keys. We propose a new group signature scheme from lattices with VLR, which achieves stronger security than the previous related works. To avoid signature forgeries, we suggest a new zero-knowledge proof system that requires signers to validate themselves. Moreover, we output an efficient tracing mechanism.

**Keywords:** lattice-based group signatures; verifier-local revocation; almost-full anonymity; traceability; zero-knowledge proof

## 1. Introduction

Group signatures, first proposed by Chaum and van Heyst [1], permit members of a group to issue signatures in the name of the group while hiding their information (anonymity). In spite of the group manager, he/she can cancel the anonymity of the signatures and identify the owner of the signature (traceability). In other words, in group signature schemes, the signature receivers can only validate the signatures, he/she cannot identify the signers. Still, in the case of dispute, an authorized person (the group manager) can recognize the signer. Thus, the signer should be anonymous to the receivers (outsiders) and traceable to the authorities (the group manager). These two features (*anonymity* and *traceability*) make group signature schemes attractive to many real-life applications, namely, key-card access systems, digital right management, and anonymous printing.

After the group signatures were proposed, many proposals were presented with several improvements and security. For instance, Chen and Pedersen [2] and Ateniese and Tsudik [3] delivered

new features such as coalition resistance, exculpability, and framing resistance. Then, Ateniese et al. [4] proposed a provably secure scheme in the random oracle model to overcome the weaknesses of the previous works. Later, Bellare et al. [5] formulated a stronger security model, namely, the BMW03 model, with two security requirements—*full anonymity* and *full traceability*, which implies the existing security properties. Even though the BMW03 model is known as the most reliable security model at present, it serves only static groups. By adopting the BMW03 model, several group signatures have been proposed, but constructing a scheme with an efficient member revocation and high-level security is a challenge.

### *1.1. Member Revocation Approaches*

In the real world, almost all group settings are stateless. Member revocation is one of the principal features of a group. Both dismissed and retired members should be restricted from generating signatures on behalf of the group in the future. One naive approach for member revocation is replacing all the keys newly except for the revoking member when he/she is revoked. Thus, any revoked member cannot produce a valid signature because he does not know the new keys. As the re-key generation approach requires the distribution of newly generated keys to all the members, verifiers, and authorities, it is not suitable for groups with large number of members. Bresson and Stern [6] proposed a method which needs signers to show that the public revocation list does not hold his member certificate when signing. Camenisch et al. [7] suggested a revocation method using dynamic accumulators. While the *accumulator* hashes a broad set of inputs to one shorter value, *dynamic accumulators* allow the insertion and deletion of inputs dynamically. The technique proposed by Camenisch et al. [7] needs existing members to store revoked user data and to update their membership for each time that a member is revoked. Thus, the proposed method is a burden for existing group members.

Brickell [8] suggested a different revocation approach, titled *verifier-local revocation (VLR)*. Afterwards, Boneh et al. [9] formalized VLR in their group signature scheme. In the VLR mechanism, every member has a revocation token, which identifies his/her status. Thus, the token of a member indicates whether he/she is revoked or not. When a user's membership is canceled, the revoking user's token is included on a list called *revocation list (RL)* and the latest revocation list is sent to the verifiers. The verifiers can use RL to authenticate the signer at the signature verification, i.e., whether the signer is an active member or not. As, in general, the number of verifiers in a group system is lower than the number of group members, the VLR mechanism is appropriate for large groups than other revocation approaches. Due to these possibilities, at the moment, VLR seems to be the most flexible revocation method.

In general, group signature schemes contain KeyGen, Sign, Verify, and Open. On the other hand, any VLR group signature scheme contains only the first three algorithms because instead of Open it has *implicit tracing algorithm* for tracing signers. The implicit tracing algorithm executes Verify for each user until it returns invalid. Then, the implicit tracing algorithm returns the index of the first user for which Verify returned invalid as the signer of the signature.

Early VLR group signature schemes were constructed on the bilinear mappings. Bilinear mappings are insecure in font of quantum computers. One of the prominent solutions for quantum attacks is lattice-based cryptography. Langlois et al. [10] proposed the first VLR group signature scheme from lattice assumptions.

### *1.2. Group Signature Schemes from Lattice Assumptions*

Cryptography based on lattice assumptions has strong security proofs in reliance on the worst-case hardness of the lattice problems. Thus, lattice cryptography seems to be an outstanding solution against quantum computers. Moreover, the efficient implementation of lattice-based cryptography attracted researches more.

Gordon et al. [11] proposed the first group signature scheme from lattice assumptions. A noticeable disadvantage of this scheme is the linear barrier, i.e., the size of the group signature increases with the number of user $N$ in a group. Thus, the size of the signatures given in the scheme in [11] is $\mathcal{O}(N)$. Then, in 2012, Camenisch et al. [12] gave a more secure and efficient scheme with an anonymous attribute token system. However, Camenisch et al.'s scheme [12] also failed to provide a solution for the issue of linear barrier. Finally, Languillaumie et al. [13] resolved the linear barrier issue in their scheme. Thus, the sizes of keys and the signatures are proportional to $\log N$ in their scheme.

However, the above-mentioned three-group signature schemes from lattices support only static groups. Later, Langlois et al. [10] suggested the first member revocable group signature scheme from lattice assumptions. The scheme in [10] employed *Verifier-local Revocation (VLR)* as the revocation mechanism. Moreover, their scheme has several advantages over previously proposed works. For instance, the scheme given in [10] is simple, as the signature of the scheme is an all-in-one proof of knowledge. Further, the scheme [10] has shorter signatures and group public keys comparing to previous schemes. Even though the scheme in [10] has several remarkable advantages over the previous works, the security of the scheme is weaker as the scheme satisfies a relaxed security notion called *selfless-anonymity*. Moreover, like any other VLR group signature scheme, Langlois's scheme employs the implicit tracing algorithms for tracing signers. The implicit tracing algorithm requires running Verify until the algorithm returns invalid. Thus, tracing a signer in a large group using the implicit tracing mechanism is not efficient.

Most of the schemes proposed after 2003 use the BMW03 model, which is known as the reliable security model at present. Among those schemes, the lattice-based group signature scheme suggested by Ling et al. [14] showed outstanding features. For instance, the size of the public key and the signature of the scheme are relatively shorter. Moreover, the scheme itself simpler because the construction of the scheme is based on Boyen's signature scheme [15]. Boyen's signature scheme has a simple construction. In addition to these advantages, the scheme in [14] has a ring variant. However, as the scheme in [14] does not support member registration or member revocation, it is suitable only for static groups.

Later, Nguyen et al. [16] also proposed a simpler group signature scheme for static groups. In their proposal, the security is reduced to the hardness of Short Integer Solution (SIS) and Learning With Errors (LWE) in the random oracle model.

Libert et al. [17] formed a group signature scheme based on lattice assumptions which facilitates member registration. Again, Ling et al. [18] delivered the first fully dynamic group signature scheme from lattices which supports both new user registration and revocation. They used accumulators to manage the member status. However, employing accumulators appears to be less efficient than employing VLR for member withdrawal in large groups. On the other hand, VLR group signatures like that in [10] failed to achieve stronger security described in the BMW03 model. Ishida et al. [19] presented an encryption mechanism to achieve full anonymity for VLR group signature schemes based on general assumptions. On the other hand, Perera et al. [20] proposed a new, stronger security notion for VLR group signature schemes.

The scheme in [20], which was constructed based on general assumptions, provides member revocation with VLR and satisfies stronger security than the security offered in the naive VLR group signature schemes. The scheme given in [20] suggested a security notion, *almost-full anonymity*. The almost-full anonymity is a controlled version of the full anonymity. In the anonymity game of almost-full anonymity, the adversary gets all the user secret signing keys, as in the full-anonymity game, and it allows revocation query as an additional feature. However, it prevents the adversary from accessing revocation tokens related to the challenging indices and generating the challenging signature for the indices, which are used in the revocation queries. In their scheme, they generated revocation tokens without depending on the secret signing keys of the members. In the VLR group signature scheme given in [10], revocation tokens are part of the particular secret signing keys. As the adversary gets all the users' secret keys at the anonymity game of the almost-full anonymity, the adversary can create revocation tokens using the information he/she has. VLR schemes become insecure when

revocation tokens are generated using secret signing keys and providing the secret keys to the adversary (as in full anonymity). Thus, they [20] delivered a different method to produce revocation tokens, which is independent of the secret signing keys. Later, employing the almost-full anonymity notion, several schemes [21–23] were proposed with different aspects. Among them, the scheme in [21] extended the almost-full anonymity notion to a new security notion to operate in fully dynamic group signature schemes with member registration and VLR. However, the scheme in [20] and its application schemes did not provide a concrete solution to secure schemes from forgery members, who replace real revocation token with a random value.

While the schemes mentioned above focused on achieving stronger security for VLR group signature schemes, the schemes provided in [24,25] tried to gain efficient revocation check at signature validation. The size of the revocation list increases dramatically when VLR schemes apply in systems where the members are joining for a short period. Long revocation lists increase the cost of authenticating the signer at the signature verification. To reduce the cost of verifying the signer, Chu et al. [24] proposed time-bound keys. Later, Perera et al. [25] employed Chu's proposal and presented a VLR group signature scheme with stronger security.

### *1.3. Our Contribution*

In this paper, we present a VLR group signature scheme from lattices with stronger security. We select the almost-full anonymity suggested in [20] to secure our scheme, as it achieves stronger security than the naive VLR group signature schemes with separate token generation. The scheme given in [20] is based on general assumptions. We use modified Boyen's signature [15] given in [14] to generate revocation token in our lattice-based group signature scheme. Moreover, we use an extra public key parameter for making revocation tokens of the users.

As the tokens are independent of the secret signing keys in the proposing scheme, there is a risk that the revoked users are generating valid signatures with arbitrary tokens. As a solution for this weakness, we present a new zero-knowledge proof system by modifying the existing zero-knowledge proof system given in [26]. Thus, the proposed scheme employs the new protocol to convince the verifiers that the signer is indeed a valid group member with a real and active token.

The implicit tracing algorithm given in VLR group signature schemes is not suitable for large groups because the time consumption is high in tracing a signer. Thus, we use the explicit tracing algorithm in our scheme to identify any signer. Accordingly, we use the group manager's secret key to find the signers instead of executing Verify a linear time in the number of users as in previous VLR group signature schemes with the implicit tracing algorithm. As the explicit-tracing algorithm helps to capture any signer only decrypting the signature, this can be used for any large groups.

As a result, we propose a group signature scheme from lattice assumptions, which is almost fully secured; supports member revocation and tracing signers efficiently, which provides a new method to generate revocation tokens; and is suitable even for a large group. Moreover, using the proposed zero-knowledge protocol, we secure our scheme from forgery signers.

### *1.4. Road Map*

In Section 2, we provide the preliminaries, and in Section 3, we discuss some of the existing security notions, the difficulty of adapting the BMW03 model to cope with revocation queries, and recall the security notion, *almost-full anonymity*. In Section 4, we provide a new zero-knowledge interactive protocol which supports the proposing scheme. In Section 5, we provide our VLR group signature scheme from lattices, including a different method for generation of revocation tokens, explicit tracing algorithm, and underlying interactive argument system. The proof of the correctness and the security of the scheme is discussed in Section 6. In Section 7, we conclude the paper and present open problems.

## 2. Preliminaries

### *2.1. Notations*

We express the set of integers $\{1, \ldots, i\}$ for any integer $i \geq 1$ by $[i]$. We declare matrices by bold upper-case letters, vectors by bold lower-case letters, and we work with only column vectors. The concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times k}$ is expressed 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 represented by $(\mathbf{x}\|\mathbf{y}) \in \mathbb{R}^{m+k}$. If $D$ is a finite set, $a \xleftarrow{\$} D$ presents that $a$ is selected uniformly at random from $D$. If $D$ is a probability distribution, $a \xleftarrow{\$} D$ indicates that $a$ is drawn according to $D$.

The security parameter of our scheme is $n$ and the maximum number of expected group users is $N = 2^\ell$. We express the binary representation of each user's index by a string as $d \in \{0,1\}^\ell$. Based on $n$, we fix the other parameters as below.

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

Let $k_1 := m + \ell$ and $k_2 := n + m + \ell$. Let $\chi$ be a $b$-bounded distribution over $\mathbb{Z}$. The norm bound for LWE noises is integer $b$ such that $q/b = \ell\tilde{\mathcal{O}}(n)$.

Let $\mathcal{H}_1$: $\{0,1\}^* \to \mathbb{Z}_q^{n \times \ell}$, $\mathcal{H}_2$: $\{0,1\}^* \to \{1,2,3\}^t$, and $\mathcal{G}$: $\{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ are hash functions, modeled as random oracles.

### *2.2. Lattices*

For integers $r, m$, prime $q \geq 2$, $\mathbf{B} = [\mathbf{b}_1| \cdots |\mathbf{b}_m] \in \mathbb{Z}_q^{r \times m}$, and $\mathbf{z} \in \mathbb{Z}^r$, the lattice $\Lambda(\mathbf{B})$ for $\mathbf{B}$ is declared as

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

where $r$ is the dimension of the lattice $\Lambda(\mathbf{B})$.

Gaussian distribution for a lattice: For a vector $\mathbf{c}$ and a parameter $s > 0$, the discrete Gaussian distribution $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2}$. The respective probability density function proportional to $\rho_{s,\mathbf{c}}$ is $D_{s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/s^n$ for all $\mathbf{x} \in \mathbb{R}^n$. The discrete Gaussian distribution with respect to a lattice $\Lambda$ is $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = D_{s,\mathbf{c}}(\mathbf{x})/D_{s,\mathbf{c}}(\Lambda) = \rho_{s,\mathbf{c}}(\mathbf{x})/\rho_{s,\mathbf{c}}(\Lambda)$ for all $\mathbf{x} \in \Lambda$. As $\mathbb{Z}^m$ is also a lattice, we can declare a discrete Gaussian distribution for $\mathbb{Z}^m$. By $D_{\mathbb{Z}^m,\sigma}$, the discrete Gaussian distribution for $\mathbb{Z}^m$ around the origin with the standard deviation $\sigma$ is expressed.

### *2.3. Lattice Hardness Assumptions*

Here, we describe the hardness of computational problems of lattices that we use in our scheme. First, we define SIVP problem. Then, we outline the two main average-case problems—LWE and SIS—and the hardness of them. We prove our scheme's security based on their hardness.

#### 2.3.1. Approximate Shortest Independent Vectors Problem ($SIVP_\gamma$)

In general, finding a good basis for a given lattice is called the *basis reduction* problem and SIVP is one of basis reduction problems.

**Definition 1** (Approximate Shortest Independent Vectors Problem -$SIVP_\gamma$ [27])**.** *Given a basis $\boldsymbol{B}$ of an $n$-dimensional lattice $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$, finding linearly independent vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ is $SIVP_\gamma$ problem, where $\|\boldsymbol{s}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$ for all $i$ ($\lambda_n(\mathcal{L})$ is $n$-th successive minimum).*

2.3.2. Learning With Errors (LWE)

Regev [28] introduced LWE problem, which is a lattice problem that is hard to solve. His work results in a reduction from worst-case lattice problems to a certain learning problem.

**Definition 2** (Learning With Errors Problem -$\mathbf{LWE}_{n,q,\chi}$ [27]). *For integers $n, m \geq 1, q \geq 2$, $\boldsymbol{s} \in \mathbb{Z}_q^n$, and $\chi$, the distribution $\boldsymbol{B}_{\boldsymbol{s},\chi}$ is achieved by sampling uniformly random $\boldsymbol{b} \in \mathbb{Z}_q^n$ and $\mathrm{e} \leftarrow \chi$, and returning the pair $(\boldsymbol{b}, \boldsymbol{b}^T \cdot \boldsymbol{s} + \mathrm{e})$.*

*Search-LWE* and *Decision-LWE* are the two versions of LWE problems. While Search-LWE is for finding the secret **s**, Decision-LWE id for distinguishing LWE samples and samples chosen according to the uniform distribution. For our scheme we employ Decision-LWE problem.

For $\beta \geq \sqrt{n}\omega(\log n)$, a prime power $q$, and distribution $\chi$, solving $LWE_{n,q,\chi}$ problem is at least as hard as solving $SIVP_\gamma$, where $\gamma = \tilde{O}(nq/\beta)$ [28,29].

2.3.3. Short Integer Solution ($SIS_{n,m,q,\beta}$)

Ajtai [30] introduced SIS in a seminal work. SIS has served in many applications as identification schemes, one-way hash functions, and digital signatures.

**Definition 3** (Short Integer Solution Problem -$\mathbf{SIS}_{n,m,q,\beta}$ [27,28]). *Given $m$ uniformly random vectors $\boldsymbol{b}_i \in \mathbb{Z}_q^n$, the columns of a matrix $\boldsymbol{B} \in \mathbb{Z}_q^{n \times m}$, SIS requires to find a nonzero vector $\boldsymbol{x} \in \Lambda^{\perp}(\boldsymbol{B})$ which forms $\|\boldsymbol{x}\| \leq \beta$ and $\boldsymbol{B}\boldsymbol{x} = 0 \mod q$.*

SIS problem is for *homogeneous* systems. Later, Gentry et al. [29] formalized its *inhomogeneous* version ISIS problem.

**Definition 4** (Inhomogeneous Short Integer Solution Problem ($\mathbf{ISIS}_{n,m,q,\beta}$) [29]). *Given matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ with $m$ uniformly random vectors $\boldsymbol{a}_i \in \mathbb{Z}_q^n$ and a uniformly random vector $\mathbf{y} \in \mathbb{Z}_q^n$, $\mathbf{ISIS}_{n,m,q,\beta}$ requires to determine a nonzero vector $\boldsymbol{x} \in \Lambda_{\boldsymbol{u}}^{\perp}(\boldsymbol{B})$ satisfying $\|\mathbf{x}\| \leq \beta$ and $\mathbf{B} \cdot \mathbf{x} = \mathbf{y} \mod q$.*

For any $m$, $\beta$, and for any $q \geq \beta \cdot \omega(\sqrt{n \log n})$, solving $SIS_{n,m,q,\beta}$ and $ISIS_{n,m,q,\beta}$ problems with non-negligible probability is at least as hard as resolving $SIVP_\gamma$ challenge, for some $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ [29].

*2.4. Lattice-Related Trapdoors*

For our construction, we require a family of functions such that each function capable of computing with any input but not feasible to invert the given input. Such a family of functions is called *one-way functions*. *Trapdoor functions* are one-way functions with secret information (trapdoor). Without this secret information, finding the inverse of the function is hard. We use trapdoor functions in our constructions as no one can identify the inverse of the function without the trapdoor.

We employ SampleD, which is a randomized nearest-plane algorithm discussed in [29,31].

- SampleD($\mathbf{T_A}$, **A**, **u**, $\sigma$): For any vector **u** in the image of **A**, a trapdoor $\mathbf{T_A}$, and $\sigma = \omega(\sqrt{n \log q \log n})$ SampleD samples $\mathbf{x} \in \mathbb{Z}^m$ from the distribution $D_{\mathbb{Z}^m,\sigma}$, such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.

The notion of preimage sampleable trapdoor functions (PSTFs) was discussed in [29]. PSTFs are defined by probabilistic polynomial-time algorithms. There are several constructions of PSTFs. We use GenTrap, SamplePre, and ExtBasis given in [13,31–33].

- GenTrap($n$, $m$, $q$): For inputs integers $n \geq 1, q \geq 2$, and $m \geq 2n \log q$ the efficient randomized algorithm GenTrap returns a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor $\mathbf{T_A}$. The distribution of **A** is negl($n$)-far from the uniform distribution.

- SamplePre($\mathbf{A}$, $\mathbf{T_A}$, $\mathbf{u}$, $\sigma$): For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a trapdoor basis $\mathbf{T_A}$, a target image $\mathbf{u} \in \mathbb{Z}_q^n$, and the standard deviation $\sigma = \omega(\sqrt{n \log q \log n})$, SamplePre samples $\mathbf{e} \in \mathbb{Z}^m$ from a distribution. $\mathbf{e}$ is within negligible statistical distance of $D_{\Lambda_q^{\mathbf{u}}(A),\sigma}$.
- ExtBasis($\mathbf{T_A}$, $\mathbf{B}$): ExtBasis gets a matrix $\mathbf{B} \in \mathbb{Z}^{n \times m'}$ and a arbitrary $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$ as inputs, where $\mathbf{A}$ is the top $n \times m$ submatrix of $\mathbf{B}$, and returns a basis $\mathbf{T_B}$ of $\Lambda_q^{\perp}(\mathbf{B})$ with $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$.

Moreover, for the construction of the underlying argument system discussed in Section 5, we employ two techniques: *witness decomposition and extension* (WitnessDE) and *matrix extension* (MatrixExt) detailed in [10].

- WitnessDE results $p$ vectors $z_1, \ldots, z_p \in \mathsf{SecretExt}(d)$ for some $d = d[1] \cdots d[\ell] \in \{0,1\}^\ell$ on input $\mathbf{x}$, where $\mathbf{x}$ is the witness of the prover $d$ and $d[i]$ is the $i$-th bit of the binary representation of $d$.

  $\mathsf{SecretExt}(d)$ is a set of all vectors $\mathbf{x} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \Sigma^{(2\ell+1)3m}$ with $2\ell+1$ blocks of size $m$, where $\ell+1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]}$ are elements of $\{-1,0,1\}^{3m}$, and remaining blocks are zero-blocks $\mathbf{0}^{3m}$.
- MatrixExt results an extended matrix $\mathbf{A}^* \in \mathbb{Z}_q^{n \times (2\ell+1)3m}$ on input matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, where $\mathbf{A}^*$ is produced by attaching *2m zero-columns* to each of the $\mathbf{A}$'s component-matrices.

### *2.5. VLR Group Signatures*

VLR group signatures are for dynamic groups. When a member is revoked, VLR group signatures require distributing the revocation list only to the verifiers. In this section, first, we provide algorithms of group signature schemes for static groups. Then, we present the algorithms of VLR group signature schemes.

Algorithms of the group signature schemes for static settings are as below.

- KeyGen($n$,$N$): This randomized PPT algorithm returns a group public key **gpk**, a group manager secret key **gmsk**, and group members' secret keys **gsk**s.
- Sign(**gpk**, **gsk**[$d$], $M$): On input **gpk**, **gsk**[$d$], and a message $M$, this randomized algorithm returns a signature $\Sigma$.
- Verify(**gpk**, $M$, $\Sigma$): This deterministic algorithm confirms that the received $\Sigma$ is valid on $M$.
- Open(**gmsk**, $M$, $\Sigma$): For given **gmsk**, $M$, and $\Sigma$ Open returns the index of the signer. If Open cannot find the signer, then it returns the failure.

VLR group signature schemes consist of three PPT algorithms [9]. It does not have an Open algorithm as it uses the *implicit tracing algorithm* to identify the corrupted users.

- KeyGen($n$,$N$): On inputs $n$ and $N$ KeyGen outputs **gpk**, a set of user secret signing keys **gsk**, and a set of user revocation tokens **grt**.
- Sign(**gpk**, **gsk**[$d$], $M$): On inputs **gpk**, **gsk**[$d$], and a message $M$, this randomized algorithm returns a signature $\Sigma$.
- Verify(**gpk**, $RL$, $M$, $\Sigma$): On inputs **gpk**, $RL$, $\Sigma$, and $M$, this algorithm confirms that the $\Sigma$ is generated on $M$ and signer's token is not in $RL$.

*Implicit tracing algorithm* employs **grt** as the tracing key. For a given valid ($M$, $\Sigma$) pair, authority, that knows all the tracing keys **grt**, can execute Verify(**gpk**, $RL$ = **grt**[$i$], $M$, $\Sigma$) for all users until Verify outputs *Invalid*. The first index that Verify returns *Invalid* is the index of the signer. The implicit tracing algorithm fails if it returns *Valid* for all the members for the input signature. In the implicit tracing algorithm to detect a single user the group manager has to check almost all users until he/she finds the signer. The time consumption of the implicit tracing algorithm is high. Thus, it is not suitable for large groups.

### *2.6. Some Other Techniques*

The underlying argument system given in Section 5 enables the signer to convince the verifier the validity of the signer, he is not being revoked, and his/her token is true.

We build our scheme based on the construction of Langloi's scheme [10]. Therefore, our scheme is based on 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 $\mathbf{u} \in \mathbb{Z}_q^n$. Each member has a revocation token to confirm their validity to sign on messages. On the other hand, the *Revocation List*, which is denoted by *RL*, contains all the revocation tokens of the revoked users. Thus, checking *RL* the verifiers can validate the signer.

*One-time signature scheme* $\mathcal{OTS}$ = (OGen, OSign, OVer): While OGen produces keys, OSign generates signature, and OVer allows the verification of the signatures [34]. Thus, OGen creates a signing/verification key pair (**osk**, **ovk**) for input $(1^n)$. On inputs **osk** and a message $M$, OSign makes a signature $\Sigma$. For given **ovk**, $M$, and $\Sigma$, OVer validates $\Sigma$ [35]. $\mathcal{OTS}$ is a one-way function. One-way functions are simpler to implement and are computationally efficient than trapdoor functions. On the other hand, $\mathcal{OTS}$ schemes are digital signature schemes. Thus it necessary to produce keys for each message. As a result, created keys are unique for the particular messages.

## 3. Definitions of the Security Notations

First, this section discusses the existing security notions for anonymity. Then, it justifies the difficulties of achieving full-anonymity for VLR schemes with revocation query and defines the almost-full anonymity. Finally, it declares the security notion traceability.

Since Chaum and van Heyst [1] introduced group signatures, more security properties have been considered according to the requirements of different applications of group signature schemes. As a result, we have a large set of security requirements including anonymity, traceability, unlinkability, unforgeability, and collusion resistance whose definitions and relations to each other have not been clearly understood [5].

Simply, anonymity and traceability can be defined as below.

- *Anonymity*: no adversary should be able to determine the index of the signer from its signature, which is produced by one of the indices from two indistinguishable indices.
- *Traceability*: no adversary should be able to produce a fake signature that cannot be traced.

Later, for static group signatures, Bellare et al. [5] formed two security standards: *full anonymity* and *full traceability* (the BMW03 model), which implies the existing unformalized requirements. In 2004, for group signatures with VLR, Boneh et al. [9] suggested a relaxed anonymity notions called *selfless-anonymity*.

### *3.1. Anonymity*

- *Full anonymity* allows the adversary to obtain secret keys of all the users and the verification key. Moreover, he/she can access the opening oracle.
- *Selfless-anonymity* does not provide user secret keys without a request, but allows *Signing*, *Corruption*, and *Revocation* queries.

The selfless-anonymity game between a challenger $C$ and an adversary $A$ is as below.

The adversary is weaker in the selfless-anonymity game than the adversary in full anonymity game because in the selfless-anonymity game the adversary cannot access all the users' secret signing keys. The adversary has to determine the key which is used to generate the signature in this game.

- **Initial Phase:** The challenger $C$ gets **gpk**, **gsk**, and **grt** from KeyGen algorithm and sends **gpk** to the adversary $A$.
- **Query Phase:** $A$ is allowed for the below three queries.

1. Signing: $A$ queries a signature for any message $M$ and an user index $i$. Then, $C$ sends back $\Sigma$ = Sign(**gpk**, **gsk**[$i$], $M$).
2. Corruption: $A$ queries the secret signing key of user $i$, and $C$ gives **gsk**[$i$].
3. Revocation: $A$ asks for the revocation token of user $i$, and $C$ sends **grt**[$i$].

- **Challenge Phase:** $A$ sends a message $M$ and two distinct identities $i_0, i_1$, such that $A$ did not make the corruption or revocation queries for $i_0, i_1$. Then, $C$ picks a bit $b \xleftarrow{\$} \{0,1\}$, produces and returns the signature $\Sigma^*$ = Sign(**gpk**, **gsk**$[i_b], M$).
- **Restricted Queries:** $A$ can do the above queries but with the below conditions.

1. Signing: $A$ is allowed to query as before.
2. Corruption: $A$ is not allowed to query for $i_0$ or $i_1$.
3. Revocation: $A$ is not allowed to query for $i_0$ or $i_1$.

- **Guessing Phase:** $A$ sends a bit $b'$ as the guess of $b$. If $b' = b$, then $A$ wins.

The advantage of $A$ winning the game is $Adv_A = |\Pr[b' = b] - 1/2|$. We say that any group signature scheme is *selfless-anonymous* if $Adv_A$ is negligible.

The first VLR group signature scheme from lattices confines on the selfless-anonymity. Our goal is to present a lattice-based VLR group signature scheme with strong security. A naive adaptation of the full anonymity (BMW03 model) does not go well since it was presented for static groups.

### *3.2. Coping with Revocation queries for Full Anonymity*

Since the *full anonymity* was originally proposed for static groups, the revocation query is not incorporated in the naive full anonymity game given in the BMW03 model or the other schemes that used the BMW03 model, such as in [14]. Our scheme is for dynamic-groups supporting member revocation. As we wish to make our VLR group signature scheme full-anonymous, we concern a security notion for "full anonymity with revocation query". On the other hand, we have to deal with the risk of giving revocation tokens to the adversary. Simply adding the revocation query given in the selfless-anonymity to the notion of the full anonymity will make our scheme insecure. The definition of the full anonymity after adding revocation query is as below.

- **Initial Phase:** The challenger $C$ gets **gpk**, a group manager's secret key **gmsk**, and group users' secret signing keys **gsk** and revocation tokens **grt**, and then delivers (**gpk**, **gsk**) to the adversary $A$.
- **Query Phase:** $A$ is allowed to request token (**grt**) of any user and opening of any signature.
- **Challenge Phase:** $A$ sends a message $M$ and two distinct identities $i_0, i_1$. Then, $C$ decides a bit $b \xleftarrow{\$} \{0,1\}$, produces and sends back $\Sigma^*$ = Sign(**gpk**, **gsk**$[i_b]$, **grt**$[i_b], M$). The adversary $A$ still is allowed to access opening oracle with any signature except the signature challenged but he/she is not allowed for revocation queries.
- **Guessing Phase:** Finally, $A$ returns a bit $b'$, the guess of $b$. If $b' = b$, then he wins.

Here, if the adversary $A$ calls the challenge phase with the indices whose revocation tokens are already queried, and if we generate the challenging signature without any restrictions, then the adversary $A$ can guess the index that used to generate the challenging signature easily. The adversary $A$ can execute Verify with all the revocation tokens he/she has, and figure the index of the generated signature. The advantage of $A$ in winning the game is $Adv_A = |\Pr[b' = b] - 1/2|$. As the adversary can obtain the tokens of the challenged indices, he/she can win the game easily. Thus, $Adv_A$ is not negligible.

In such a way, allowing the adversary $A$ to query any revocation token and generating the challenged signature for the indices, even those indices' revocation tokens are queried, makes the scheme non-secured.

Because of this problem, we have to consider a security notion which has the restrictions of providing revocation tokens to the adversary. Thus, we use the almost-full anonymity given in [20], which is a restricted version of full anonymity.

### *3.3. Almost Full Anonymity*

The idea of *almost full anonymity* is depicted in Figure 1. Here, as the naive full anonymity game, the challenger $C$ creates the keys, and gives **gpk** and **gsk** to the adversary $A$. When $A$ accesses the opening oracle with a message–signature pair $(M, \Sigma)$, the oracle outputs $\mathsf{Open}(\mathbf{gmsk}, M, \Sigma)$ as usual. Furthermore, $A$ can request the token of any user $d$. Thus, $C$ replies with $\mathbf{grt}[d]$. The revocation query is not in the original notion of full anonymity. Then, $A$ sends two valid identities $i_0, i_1$ with a message $M$. Then, $C$ chooses one of the two identities, which are not being queried before in revocation query phase, and outputs the signature $\Sigma^*$. Here, signatures are not generated for the indices that have been queried for revocation tokens as the adversary $A$ can use the tokens and execute Verify to check the generated signature. The goal of $A$ is to recognize the index that is employed to produce $\Sigma^*$. He/she is still allowed to access the opening oracle, but without the challenged signature, and he/she can request revocation token of any user except the challenging indices. The almost-full anonymity game between a challenger $C$ and an adversary $A$ is as below.

- **Initial Phase:** $C$ runs the algorithm KeyGen and gets **gpk**, **gmsk**, and group members' secret keys **gsk** and revocation tokens **grt**, and then sends (**gpk**, **gsk**) to $A$.
- **Query Phase:** $A$ can ask token of any user, and request the opening of any valid $(M, \Sigma)$ pair.
- **Challenge Phase:** $A$ sends a message $M$ and two distinct indexes $i_0, i_1$, such that $A$ never queried the tokens of them. $C$ chooses a bit $b \xleftarrow{\$} \{0,1\}$, creates and sends back $\Sigma^* = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i_b], \mathbf{grt}[i_b], M)$.

  The adversary $A$ still can query the opening oracle without the signature challenged, and he/she can request revocation tokens of any user except the indices used for challenge.
- **Guessing Phase:** Finally, $A$ sends a bit $b'$, the guess of $b$. If $b' = b$, then he wins.

We declare the advantage of $A$ in the above game as $Adv_A = |\Pr[b' = b] - 1/2|$. We say that any group signature is *almost-full anonymous* if for all polynomial $N$ and for all adversaries, the $Adv_A$ is negligible in the security parameter $n$.

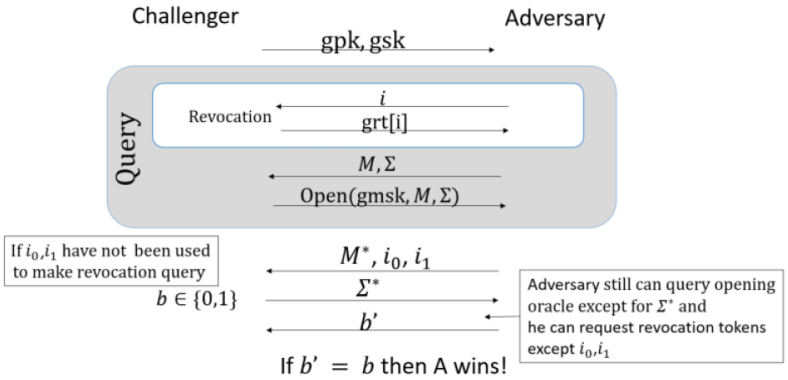


**Figure 1.** Almost-full anonymity.

Here, we discuss the almost-full anonymity with regards to the full anonymity and the selfless-anonymity. As in any other anonymity game, in the almost-full anonymity game, **gpk** is given to the adversary $A$ and, as in the full-anonymity, all the user secret signing keys **gsk** are given to the adversary $A$ at the beginning of the game. Even the member revocation tokens are generated; they are not provided to $A$ in the initial phase. In the query phase, $A$ can access Open as in the

full-anonymity and request for revocation tokens as in the selfless-anonymity game. Then, $A$ can output $i_0, i_1$, which are not used in the revocation query as in the selfless-anonymity. Still, $A$ can access Open but not with the signature challenged, and he/she is permitted for further revocation queries except for indices challenged. Thus, the almost-full anonymity is stronger than the selfless-anonymity as all the users' secret signing keys are provided to the adversary. However, the almost-full anonymity is not as strong as the full anonymity because we cannot permit the adversary to access all the revocation tokens. However, all the users' secret signing keys are provided to the adversary. In the full anonymity given in the BMW03 model, all the secret signing keys of the users (the only secret key of the users in that scheme has) are provided to the adversary. However, in the VLR scheme, we have another user's key called tracing key (revocation token) which cannot disclose to the adversary without any restrictions. Thus we say the almost-full anonymity is a controlled variant of the full anonymity, and it is somewhat weaker than the full anonymity. A scheme to be fully anonymous, all the secret keys (both secret signing keys and revocation tokens) should be given to the adversary at the start of the game. The almost-full anonymity is a reasonable solution for our scheme rather than the selfless-anonymity. An attacker can identify the signer only if he gets hold of the signer's token. An attacker obtaining the exact signer' token is as rare.

The VLR group signature scheme in [10], generates revocation tokens **grt** by taking a part of the secret keys **gsk**. As we are providing all the users' secret signing keys to the adversary, and he/she can query revocation tokens, he/she can create challenged indices' tokens utilizing the secret keys he has. Thus, we take a different way to generate the revocation tokens, as discussed in Section 5. Thus, revocation tokens of our scheme are not derived from the secret signing keys.

### *3.4. Traceability*

The naive definition of traceability in [1] is to determine the correctness of the opening algorithm. Therefore, for a valid signature signed by $i$ with **gsk**[$i$], the opening algorithm should return $i$. Later, traceability appeared with an actual security requirement, that it is not able to create a signature which can not be locate to a group that generated the signature. However, the BMW03 model gave a much stronger notion called *full-traceability*, which can be regarded as a strong form of traceability and collusion resistance.

In the traceability game, the challenge of the adversary is to form a signature that cannot be traced. Any group signature scheme is traceable if no adversary can win this challenge. Therefore, we say that a VLR group signature scheme is traceable if the adversary cannot forge a signature that can be traced to one of the users in his coalition using the implicit tracing algorithm. The traceability game between a challenger $C$ and an adversary $A$ [10] is as follows.

- **Initial Phase:** $C$ obtains **gpk**, **gsk**, and **grt**. Then, $C$ sends (**gpk**, **grt**) to $A$ and sets corruption list $U \leftarrow \emptyset$.
- **Query Phase:** $A$ is allowed for the below queries.
    1. Signing: $A$ requests a signature by sending a message $M$ and a user index $i$, and $C$ responds with $\Sigma = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], M)$.
    2. Corruption: $A$ queries for the secret key of any user $i$. The challenger $C$ sends **gsk**[$i$] after adding $i$ to $U$.
- **Challenge Phase:** $A$ sends a message $M^*$, a set of revocation tokens $RL^*$, and a signature $\Sigma^*$.
- The forgery $A$ wins if the followings are correct.
    1. $\Sigma^*$ is valid on $M^*$ with $RL^*$.
    2. $\Sigma^*$ traces to some id outside the coalition $U \backslash RL^*$ or tracing algorithm returns failure symbol.
    3. $\Sigma^*$ is not gained by signing on $M^*$.

The advantage of $A$ is $Adv_A^{trace} = |\Pr[\mathbf{Exp}_A^{trace}(n, N) = 1]|$, where $\mathbf{Exp}_A^{trace}$ is the traceability game between the challenger $C$ and the adversary $A$. We say that a group signature scheme is traceable if $Adv_A^{trace}$ is negligible.

## 4. The Underlying Zero Knowledge Interactive Protocol

In the proposing scheme, as the tokens are separated from the secret signing keys, there is a risk of signers cheating their tokens at the time of signing. For instance, a revoked member can pick a random value as his/her token and generate a signature. As the fake token is not in the revocation list RL, the verifier thinks the signer is an active member. To prevent such forgeries, we require signers to prove that their token is valid. In other words, the signer should prove that for generating the signature he/she used token which is given at the setup stage. As the signer cannot show his token to outsiders, he/she needs a system to generate a proof while hiding his/her private data. Thus, we modify the zero-knowledge protocol given in [26], and provide a new zero-knowledge protocol that can satisfy the proposing scheme's requirements. Consequently, we present a new protocol as our scheme's underlying zero-knowledge interactive protocol that proves the signer is valid, his/her token is not revoked, and his/her token is real.

*Zero Knowledge Interactive Protocol* enables a prover (signer) to prove that he/she is a approved group member who has valid secret keys.

Let COM be the statistically hiding and computationally binding commitment scheme given in [36].

A combined interactive protocol is given in [26]. By using that protocol, a signer can prove the validity of signing, that his/her revocation token is not in the revocation list, and that his/her index is correctly encrypted. However, as we need the signer to prove that his token is real, which cannot be achieved by directly using the method in [26], we need to modify the protocol in [26].

When the protocol in [26] is used in a scheme which supports both VLR and the explicit tracing mechanism, we use public parameters, a matrix $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{V} \in \mathbb{Z}_q^{m\times n}$, a vector $\mathbf{v} \in \mathbb{Z}_q^m$, a matrix $\mathbf{P} \in \mathbb{Z}_q^{k_1\times k_2}$, and another vector $\mathbf{c} \in \mathbb{Z}_q^{k_1}$. The witness of the prover consists of a vector $\mathbf{x}^{(d)} = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \Sigma^{(2\ell+1)m}$ for some $d \in \{0,1\}^\ell$, a vector $\mathbf{e}_1 \in \mathbb{Z}^m$, a vector $\mathbf{r} \in \mathbb{Z}_q^n$, and another vector $\mathbf{e} \in \mathbb{Z}^{k_2}$. The prover has to show the verifier that $\mathbf{A}\cdot\mathbf{x} = \mathbf{u} \bmod q$, $\|\mathbf{e}_1\|_\infty \le \beta$ and $\mathbf{V}\cdot(\mathbf{B}\cdot\mathbf{r}) + \mathbf{e}_1 = \mathbf{v} \bmod q$ while keeping his identity $d$ in secret. Moreover, the prover has to show his identity is correctly encrypted, such that $\|\mathbf{e}\|_\infty \le b$ and $\mathbf{Pe} + (0^{k_1-\ell}\|\lfloor q/2\rfloor d) = \mathbf{c} \bmod q$.

In our scheme, we use the modified Boyen's signature given in [14] to make revocation tokens. In Boyen's signature schemes, it requires to show the verifier that $\mathbf{A}_{(d)}\mathbf{z} = \mathbf{u} \bmod q$, while hiding both $d$ and $\mathbf{z}$. The matrix $\mathbf{A}_{(d)}$ is unique to the users as it is obtained by using user's index. The vector $\mathbf{z}$ is attained by using $\mathbf{A}_{(d)}$. We use the same method in our scheme to prove the validity of the revocation token. Thus, in our scheme, we use $(\mathbf{A}_{(d)}\cdot\mathbf{t})$ as the revocation token of the user and as a result, in addition to the proof given in [26], we should prove that $\mathbf{A}_{(d)}\cdot\mathbf{t} = \mathbf{w} \bmod q$ in zero-knowledge, while keeping both $d$ and $\mathbf{t}$ in secret. Here $\mathbf{w}$ is a public parameter and $\mathbf{t}$ is generated using $\mathbf{A}_{(d)}$ and $\mathbf{w}$.

Using the method given in [14], we take following steps to prove $\mathbf{A}_{(d)}\cdot\mathbf{t} = \mathbf{w} \bmod q$, where $\mathbf{A}_{(d)} = [\mathbf{A}_0|\sum_{i=1}^{\ell}\mathbf{A}_i^{d[i]}\cdot d[i]]$.

Let $\bar{\mathbf{A}} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(1+\ell)m}$ and $\mathbf{t} = (\mathbf{x}\|\mathbf{y}) \in \mathbb{Z}^m$. Thus, we get

$\mathbf{w} = \mathbf{A}_{(d)}\cdot\mathbf{t} = \mathbf{A}_0\mathbf{x} + \sum_{i=1}^{\ell}\mathbf{A}_i^{d[i]}\cdot(d[i]\mathbf{y}) = \bar{\mathbf{A}}\bar{\mathbf{t}} \bmod q$, where $\bar{\mathbf{t}} = (\mathbf{x}\|d_1\mathbf{y}\|\ldots\|d_\ell\mathbf{y})$.

We need to argue the position of $\bar{\mathbf{t}} \in \mathbb{Z}^{(1+\ell)m}$ in zero-knowledge, for a given $(\bar{\mathbf{A}}, \mathbf{w})$, such that $\|\bar{\mathbf{t}}\|_\infty \le \beta$ and $\bar{\mathbf{A}}\bar{\mathbf{t}} = \mathbf{w} \bmod q$.

Let $\mathbf{A}^* = [\mathbf{A}_0|0^{n\times 2m}|\mathbf{A}_1^0|0^{n\times 2m}|\mathbf{A}_1^1|0^{n\times 2m}|\ldots|\mathbf{A}_\ell^0|0^{n\times 2m}|\mathbf{A}_\ell^1|0^{n\times 2m}|0^{n\times 3m\ell}] \in \mathbb{Z}_q^{n\times(1+2\ell)3m}$ and $\mathbf{t}_j = (\mathbf{x}_j\|d_1\mathbf{y}_j\|\ldots\|d_{\ell+1}\mathbf{y}_j\|\ldots\|d_{2\ell}\mathbf{y}_j) \in \{-1,0,1\}^{(1+2\ell)3m}$.

$\mathbf{A}^*$ is the extended matrix [10] of $\bar{\mathbf{A}}$, and $\mathbf{t}_j$ is the elementary extension [10] of $\bar{\mathbf{t}}$.

We can argue $\mathbf{A}^*(\sum_{j=1}^{p}\beta_j\mathbf{t}_j) = \mathbf{w} \bmod q$.

Using masking vectors $(\mathbf{r}_t^{(1)}, \ldots, \mathbf{r}_t^{(p)}) \in \mathbb{Z}_q^{n\times(1+2\ell)3m}$ instead of arguing above we can show, $\mathbf{A}^* \sum_{j=1}^{P} \beta_j(\mathbf{t}_j + \mathbf{r}_t^{(j)}) - \mathbf{w} = \mathbf{A}^*(\sum_{j=1}^{p} \beta_j \mathbf{r}_t^{(j)}) \mod q$.

In our new zero-knowledge protocol the public inputs are a vector $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$, vector $\mathbf{u} \in \mathbb{Z}_q^n$, vector $\mathbf{w} \in \mathbb{Z}_q^n$, matrix $\mathbf{V} \in \mathbb{Z}_q^{m\times n}$, vector $\mathbf{v} \in \mathbb{Z}_q^m$, matrix $\mathbf{P} \in \mathbb{Z}_q^{k_1\times k_2}$, and a vector $\mathbf{c} \in \mathbb{Z}_q^{k_1}$. The witness of the prover consists of a vector $\mathbf{x}^{(d)} = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \Sigma^{(2\ell+1)m}$ for some $d \in \{0,1\}^\ell$, a vector $\mathbf{e}_1 \in \mathbb{Z}^m$, a vector $\mathbf{t} \in \mathbb{Z}_q^{2m}$, and another vector $\mathbf{e} \in \mathbb{Z}^{k_2}$. The prover should persuade the verifier that $\mathbf{A}\cdot\mathbf{x} = \mathbf{u} \mod q$, $\|\mathbf{e}_1\|_\infty \le \beta$ and $\mathbf{V}\cdot(\mathbf{A}_{(d)}\cdot\mathbf{t}) + \mathbf{e}_1 = \mathbf{v} \mod q$ while hiding prover's identity $d$. Moreover, the prover has to show that his/her identity is correctly encrypted, such that $\|\mathbf{e}\|_\infty \le b$ and $\mathbf{Pe} + (0^{k_1-\ell}\|\lfloor q/2\rfloor d) = \mathbf{c} \mod q$ and token is real, such that $\mathbf{A}_{(d)}\cdot\mathbf{t} = \mathbf{w} \mod q$.

### *4.1. Preparation Step*

- The common inputs: Matrices $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$, $\mathbf{V} \in \mathbb{Z}_q^{m\times n}$, and $\mathbf{P} \in \mathbb{Z}_q^{k_1\times k_2}$ and vectors $\mathbf{u}, \mathbf{w} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{v} \in \mathbb{Z}_q^m$, and $\mathbf{c} \in \mathbb{Z}_q^{k_1}$.
- The prover's inputs: A vector $\mathbf{x} = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \mathsf{Secret}_\beta(d)$ for some secret $d \in \{0,1\}^\ell$, vector $\mathbf{e}_1 \in \mathbb{Z}^m$, vector $\mathbf{t} \in \mathbb{Z}_q^{2m}$, and a vector $\mathbf{e} \in \mathbb{Z}^{k_2}$. We use $\mathbf{f}$ instead of $\mathbf{e}_1$ hereunder to discard the confusing $\mathbf{e}_1$with $\mathbf{e}$.
- The prover's target is to prove the verifier in zero-knowledge that
  - $\mathbf{A}\cdot\mathbf{x} = \mathbf{u} \mod q$ and $\mathbf{x} \in \mathsf{Secret}_\beta(d)$.
  - $\|\mathbf{f}\|_\infty \le \beta$ and $\mathbf{V}\cdot(\mathbf{A}_{(d)}\cdot\mathbf{t}) + \mathbf{f} = \mathbf{v} \mod q$.
  - $\|\mathbf{e}\|_\infty \le b$ and $\mathbf{Pe} + (0^{k_1-\ell}\|\lfloor q/2\rfloor d) = \mathbf{c} \mod q$ ($b$ is the norm bound for LWE noises and $\bar{p} = \lfloor \log b \rfloor + 1$).
  - $\mathbf{A}_{(d)}\cdot\mathbf{t} = \mathbf{w} \mod q$.

Before the interaction, both the prover and the verifier form the public matrices: $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$, $\mathbf{V}^* = \mathbf{V}\cdot\mathbf{A} \in \mathbb{Z}_q^{m\times m}$, $\mathbf{I}^* \in \{0,1\}^{m\times 3m}$ ($\mathbf{I}^*$ is gained by appending *2m zero-columns* to the identity matrix of order $m$), $\mathbf{P}^* = [\mathbf{P} \mid 0^{k_1\times 2k_2}] \in \mathbb{Z}_q^{k_1\times 3k_2}$, and

$$\boldsymbol{Q} = \left(\begin{array}{c|c} 0^{(k_1-\ell)\times\ell} & 0^{(k_1-\ell)\times\ell} \\ \hline \lfloor q/2\rfloor\mathbf{I}_\ell & 0^{\ell\times\ell} \end{array}\right) \in \{0, \lfloor q/2\rfloor\}^{k_1\times 2\ell}.$$

Then, the prover uses the Decomposition-Extension technique provided in [10] with his/her witness vectors as below.

- Let $\mathbf{z}_1, \ldots, \mathbf{z}_p \leftarrow \mathsf{WitnessDE}(\mathbf{x})$.
- Let $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_p \leftarrow \mathsf{EleDec}(\mathbf{f})$, then for each $i \in [p]$, let $\mathbf{f}_i \leftarrow \mathsf{EleExt}(\tilde{\mathbf{f}}_i)$.
- Let $\tilde{\mathbf{e}}_1, \ldots, \tilde{\mathbf{e}}_{\bar{p}} \leftarrow \mathsf{EleDec}(\mathbf{e})$, then for each $i \in [p]$, let $\mathbf{e}_i \leftarrow \mathsf{EleExt}(\tilde{\mathbf{e}}_i)$.
- Let $\tilde{\mathbf{t}}_1, \ldots, \tilde{\mathbf{t}}_p \leftarrow \mathsf{EleDec}(\mathbf{t})$, then for each $i \in [p]$, let $\mathbf{t}_i \leftarrow \mathsf{EleExt}(\tilde{\mathbf{t}}_i)$.

At the interactive protocol, the prover instead convince the verifier that he/she knows $\mathbf{z}_1, \ldots, \mathbf{z}_p \in \mathsf{Secret}_\beta(d)$, $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_p \in \mathsf{B}_{3m}$, $\tilde{\mathbf{t}}_1, \ldots, \tilde{\mathbf{t}}_p \in \mathsf{B}_{3m}$, and $\tilde{\mathbf{e}}_1, \ldots, \tilde{\mathbf{e}}_p \in \mathsf{B}_{3k_2}$, such that

$$\begin{cases} \mathbf{A}^*\cdot(\sum_{j=1}^{p}\beta_j\cdot\mathbf{z}_j) = \mathbf{u} \mod q; \\ \mathbf{V}^*\cdot(\sum_{j=1}^{p}\beta_j\cdot\mathbf{t}_j) + \mathbf{I}^*\cdot(\sum_{j=1}^{P}\beta_j\cdot\mathbf{f}_j) = \mathbf{v} \mod q. \\ \mathbf{P}^*\cdot(\sum_{j=1}^{\bar{p}} b_j\cdot\mathbf{e}_j) + \mathbf{Q}\cdot d^* = \mathbf{Pe} + (0^{k_1-\ell}||\lfloor q/2\rfloor d) = \mathbf{c} \mod q. \\ \mathbf{A}^*\cdot(\sum_{j=1}^{p}\beta_j\cdot\mathbf{t}_j) = \mathbf{w} \mod q; \end{cases}$$

*4.2. Description of the Protocol*

1. **Commitment**: The prover samples randomness $\rho_1, \rho_2, \rho_3$ for COM and the below uniformly random objects:

$$\begin{cases} c \xleftarrow{\$} \{0,1\}^{\ell}; \\ \pi_{z,1}, \ldots, \pi_{z,p} \xleftarrow{\$} S; \pi_{f,1}, \ldots, \pi_{f,p} \xleftarrow{\$} S_{3m}; \pi_{t,1}, \ldots, \pi_{t,p} \xleftarrow{\$} S_{3m}; \\ \quad \pi_{e,1}, \ldots, \pi_{e,\bar{p}} \xleftarrow{\$} S_{3k_2}; \tau \xleftarrow{\$} S_{2\ell}; \\ \mathbf{r}_{z,1}, \ldots, \mathbf{r}_{z,p} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)3m}; \mathbf{r}_{f,1}, \ldots, \mathbf{r}_{f,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \\ \quad \mathbf{r}_{t,1}, \ldots, \mathbf{r}_{t,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{r}_{e,1}, \ldots, \mathbf{r}_{e,\bar{p}} \xleftarrow{\$} \mathbb{Z}_q^{3k_2}; \mathbf{r}_d \xleftarrow{\$} \mathbb{Z}_q^{2\ell}. \end{cases} \tag{1}$$

Then, the prover sends the following commitment $\mathbf{CMT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ to the verifier.

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{t,j}\}_{j=1}^{p}), \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{z,j}); \\ \quad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{t,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{f,j}); \\ \quad \{\pi_{e,j},\}_{j=1}^{\bar{p}}; \mathbf{P}^* (\sum_{j=1}^{\bar{p}} b_j \mathbf{r}_{e,j}) + \mathbf{Q}\mathbf{r}_d; \tau; \rho_1); \\ \quad \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{t,j}), \\ \mathbf{c}_2 = \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{r}_{z,j}), \pi_{f,j}(\mathbf{r}_{f,j}), \pi_{t,j}(\mathbf{r}_{t,j})\}_{j=1}^{p}; \\ \quad \{\pi_{e,j}(\mathbf{r}_{e,j})\}_{j=1}^{\bar{p}}; \tau(\mathbf{r}_d); \rho_2), \\ \mathbf{c}_3 = \\ \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{z}_j + \mathbf{r}_{z,j}), \pi_{f,j}(\mathbf{f}_j + \mathbf{r}_{f,j}), \pi_{t,j}(\mathbf{t}_j + \mathbf{r}_{t,j})\}_{j=1}^{p}; \\ \quad \{\pi_{e,j}(\mathbf{e}_j + \mathbf{r}_{e,j})\}_{j=1}^{\bar{p}}; \tau(d^* + \mathbf{r}_d); \rho_3). \end{cases} \tag{2}$$

2. **Challenge:** The verifier sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$.
3. **Response:** Based on the challenge, the prover computes and outputs the response RSP as below.

- Case $Ch = 1$: Let $d_1 = d \oplus c$. For each $j \in [p]$, let $\mathbf{a}_{z,j} = T_c \circ \pi_{z,j}(\mathbf{z}_j); \mathbf{b}_{z,j} = T_c \circ \pi_{z,j}(\mathbf{r}_{z,j}); \mathbf{a}_{f,j} = \pi_{f,j}(\mathbf{f}_j); \mathbf{b}_{f,j} = \pi_{f,j}(\mathbf{r}_{f,j}); \mathbf{a}_{t,j} = \pi_{t,j}(\mathbf{t}_j); \mathbf{b}_{t,j} = \pi_{t,j}(\mathbf{r}_{t,j})$. For each $j \in [\bar{p}]$, let $\mathbf{a}_{e,j} = \pi_{e,j}(\mathbf{e}_j); \mathbf{b}_{e,j} = \pi_{e,j}(\mathbf{r}_{e,j})$. Let $\mathbf{a}_d = \tau(d^*); \mathbf{b}_d = \tau(\mathbf{r}_d)$. Then, send

$$RSP = (d_1, \{\mathbf{a}_{z,j}, \mathbf{b}_{z,j}, \mathbf{a}_{f,j}, \mathbf{b}_{f,j}, \mathbf{a}_{t,j}, \mathbf{b}_{t,j}\}_{j=1}^{p}, \{\mathbf{a}_{e,j}, \mathbf{b}_{e,j}\}_{j=1}^{\bar{p}}, \mathbf{a}_d, \mathbf{b}_d, \rho_2, \rho_3). \tag{3}$$

- Case $Ch = 2$: Let $d_2 = c$. For each $j \in [p]$, let $\phi_{z,j} = \pi_{z,j}; \phi_{f,j} = \pi_{f,j}; \phi_{t,j} = \pi_{t,j}; \mathbf{s}_{z,j} = \mathbf{z}_j + \mathbf{r}_{z,j}; \mathbf{s}_{f,j} = \mathbf{f}_j + \mathbf{r}_{f,j}; \mathbf{s}_{t,j} = \mathbf{t}_j + \mathbf{r}_{t,j}$. For each $j \in [\bar{p}]$, let $\phi_{e,j} = \pi_{e,j}; \mathbf{s}_{e,j} = \mathbf{e}_j + \mathbf{r}_{e,j}$. Let $\hat{\tau} = \tau$ and $\mathbf{s}_d = d^* + \mathbf{r}_d$. Then, send

$$RSP = (d_2, \{\phi_{z,j}, \phi_{f,j}, \phi_{t,j}, \mathbf{s}_{z,j}, \mathbf{s}_{f,j}, \mathbf{s}_{t,j}\}_{j=1}^{p}, \{\phi_{e,j}, \mathbf{s}_{e,j}\}_{j=1}^{\bar{p}}, \hat{\tau}, \mathbf{s}_d, \rho_1, \rho_3) \tag{4}$$

- Case $Ch = 3$: Let $d_3 = c$. For each $j \in [p]$, let $\psi_{z,j} = \pi_{z,j}; \psi_{f,j} = \pi_{f,j}; \psi_{t,j} = \pi_{t,j}; \mathbf{h}_{z,j} = \mathbf{r}_{z,j}; \mathbf{h}_{f,j} = \mathbf{r}_{f,j}; \mathbf{h}_{t,j} = \mathbf{r}_{t,j}$. For each $j \in [\bar{p}]$, let $\psi_{e,j} = \pi_{e,j}; \mathbf{h}_{e,j} = \mathbf{r}_{e,j}$. Let $\tilde{\tau} = \tau$ and $\mathbf{h}_d = \mathbf{r}_d$. Then, send

$$RSP = (d_3, \{\psi_{z,j}, \psi_{f,j}, \psi_{t,j}, \mathbf{h}_{z,j}, \mathbf{h}_{f,j}, \mathbf{h}_{t,j}\}_{j=1}^{p}, \{\psi_{e,j}, \mathbf{h}_{e,j}\}_{j=1}^{\bar{p}}, \tilde{\tau}, \mathbf{h}_d, \rho_1, \rho_2) \quad (5)$$

4. The verifier verifies received RSP as below.

- $Ch = 1$: Parse RSP as in (3).

  Check whether $\forall \in [p] : \mathbf{a}_{z,j} \in \mathsf{SecretExt}(d_1), \mathbf{a}_{f,j} \in \mathsf{B}_{3m}, \mathbf{a}_{t,j} \in \mathsf{B}_{3m}, \forall j \in [\bar{p}] : \mathbf{a}_d \in \mathsf{B}_{2\ell}, \mathbf{a}_{e,j} \in \mathsf{B}_{3k_2}$, and

$$\begin{cases} \mathbf{c}_2 = \mathsf{COM}(\{\mathbf{b}_{z,j}, \mathbf{b}_{f,j}, \mathbf{b}_{t,j}\}_{j=1}^{p}; \{\mathbf{b}_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{b}_d; \rho_2), \\ \mathbf{c}_3 = \mathsf{COM}(\{\mathbf{a}_{z,j} + \mathbf{b}_{z,j}, \mathbf{a}_{f,j} + \mathbf{b}_{f,j}, \mathbf{a}_{t,j} + \mathbf{b}_{t,j}\}_{j=1}^{p}; \\ \quad \{\mathbf{a}_{e,j} + \mathbf{b}_{e,j}\}_{j=1}^{\bar{p}}; \{\mathbf{a}_d + \mathbf{b}_d\}; \rho_3). \end{cases} \quad (6)$$

- $Ch = 2$: Parse RSP as in (4). Check whether

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}(d_2, \{\phi_{z,j}, \phi_{f,j}, \phi_{t,j}\}_{j=1}^{p}, \mathbf{A}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{z,j}) - \mathbf{u}; \\ \quad \mathbf{V}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{t,j}) + \mathbf{I}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{f,j}) - \mathbf{v}; \\ \quad \{\phi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{s}_{e,j}) + \mathbf{Q}\mathbf{s}_d - \mathbf{c}; \hat{\tau}; \rho_1); \\ \quad \mathbf{A}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{t,j}) - \mathbf{w}, \\ \mathbf{c}_3 = \mathsf{COM}(\{T_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}), \phi_{f,j}(\mathbf{s}_{f,j}), \phi_{t,j}(\mathbf{s}_{t,j})\}_{j=1}^{p}; \\ \quad \{\phi_{e,j}(\mathbf{s}_{e,j})\}_{j=1}^{\bar{p}}; \hat{\tau}(\mathbf{s}_d); \rho_3). \end{cases} \quad (7)$$

- $Ch = 3$: Parse RSP as in (5). Check whether

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}(d_3, \{\psi_{z,j}, \psi_{f,j}, \psi_{t,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{z,j}); \\ \quad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{t,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{f,j}); \\ \quad \{\phi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{h}_{e,j}) + \mathbf{Q}\mathbf{h}_d; \tilde{\tau}; \rho_1); \\ \quad \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{t,j}), \\ \mathbf{c}_2 = \mathsf{COM}(\{T_{d_3} \circ \psi_{z,j}(\mathbf{h}_{z,j}), \psi_{f,j}(\mathbf{h}_{f,j}), \psi_{t,j}(\mathbf{h}_{t,j})\}_{j=1}^{p}; \\ \quad \{\psi_{e,j}(\mathbf{h}_{e,j}), \}_{j=1}^{\bar{p}}; \tilde{\tau}(\mathbf{h}_d); \rho_2). \end{cases} \quad (8)$$

If and only if all the conditions hold, the verifier returns Valid. Otherwise, he/she returns Invalid.

As discussed in [26], we construct an efficient simulator $S$ interacting with a (probably dishonest) verifier $\hat{\mathcal{V}}$, such that, only using the public input, $S$ returns a simulated transcript that is statistically close to the one created in the real interaction by the honest prover. Thus, the simulator can successfully imitate the honest prover with probability negligibly far from 2/3.

We provide the analysis of the new underlying zero-knowledge protocol in Appendix A.

## 5. Our VLR Group Signature Scheme

In our scheme, we generate member revocation tokens using modified Boyen's signature as in [14]. Even when there is no direct relationship to the secret keys we obtain the revocation tokens by using the member-indices. Thus, each revocation token has a relation to each member's index, which is unique to the members. According to the scheme described in [10], revocation tokens can be obtained by a part of the public key and a part of the secret key. However, as we are giving all the secret keys to

the adversary in the anonymity game, the adversary may construct the challenged indices' revocation tokens by studying the pattern of the queried revocation tokens and using the secret signing keys he has. Thus, we come with a solution to obtain revocation tokens in our scheme by using modified Boyen's signature.

Our scheme consists of four algorithms as below.

- KeyGen($n$,$N$): On inputs, the security parameter $n \in \mathbb{N}$ and the number of group users $N$, this randomized PPT algorithm outputs a group public key **gpk**, a group manager secret key **gmsk**, a set of user secret keys **gsk**, and a set of user revocation tokens **grt**.
- Sign(**gpk**, **gsk**[$d$], **grt**[$d$], $M$): This algorithm computes a group signature $\Sigma$ on given $M$ using **gpk**, signer's secret signing key **gsk**[$d$], and the token **grt**[$d$].
- Verify(**gpk**, $RL$, $M$, $\Sigma$): This algorithm determines whether the given $\Sigma$ is a valid on $M$ by employing **gpk**. Then, it confirms that the signer is active by employing $RL$.
- Open(**gmsk**, $M$, $\Sigma$): This algorithm gets the group manager's secret key **gmsk**, a message–signature pair ($M$, $\Sigma$) as inputs and outputs the index of the signer. It returns failure symbol when the user cannot be identified.

### *5.1. Description of Our Scheme*

In this section, we present the construction of our scheme from lattices.

**Key Generation:** KeyGen($n$,$N$) creates a group public key **gpk**, a group manager secret key **gmsk**, group users secret signing keys **gsk**, and group user tokens **grt** as below.

1. Run $\mathsf{GenTrap}(n, m, q)$ to obtain a matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T}_\mathbf{A}$.
2. Sample vectors $\mathbf{u}, \mathbf{w} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. Sample matrices $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ 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. Execute algorithm $\mathsf{GenTrap}(n, m, q)$ from [29] to get a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T}_\mathbf{B}$.
6. For each user $d \in \{0, 1, \cdots, N-1\}$, produce secret signing keys **gsk**[$d$] and revocation tokens **grt**[$d$] as below.

   (a) Let $d[1]\cdots d[\ell] \in \{0,1\}^\ell$ be the binary representation of $d$.
   (b) Sample vectors $\mathbf{x}_1^{d[1]}\cdots\mathbf{x}_\ell^{d[\ell]} \leftarrow D_{\mathbb{Z}^m,\sigma}$.
   (c) Compute $\mathbf{z} = \sum_{i=1}^{\ell}\mathbf{A}_i^{d[i]}\cdot\mathbf{x}_i^{d[i]} \bmod q$.
   (d) Get $\mathbf{x}_0 \in \mathbb{Z}^m \leftarrow \mathsf{SampleD}(\mathbf{T}_\mathbf{A}, \mathbf{A}_0, \mathbf{u}-\mathbf{z}, \sigma)$.
   (e) Let $\mathbf{x}_1^{1-d[1]}\cdots\mathbf{x}_\ell^{1-d[\ell]}$ be zero vectors $\mathbf{0}^m$.
   (f) Define $\mathbf{x}^{(d)} = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \Sigma^{(2\ell+1)m}$.
   If $\|\mathbf{x}^{(d)}\|_\infty \le \beta$, then continue. Otherwise, redo from (a).
   (g) The user secret signing key is **gsk**[$d$] = $\mathbf{x}^{(d)}$.
   (h) Compute $\mathbf{A}_{(d)} = [\mathbf{A}_0|\sum_{i=1}^{\ell} d_i\mathbf{A}_i^{d[i]}] \in \mathbb{Z}_q^{n\times 2m}$.
   (i) Run $\mathsf{ExtBasis}(\mathbf{T}_\mathbf{A}, \mathbf{A}_{(d)})$ to obtain a short basis $\mathbf{T}_{(d)}$.
   (j) Get $\mathbf{t}^{(d)} \in \mathbb{Z}^{2m} \leftarrow \mathsf{SampleD}(\mathbf{T}_{(d)}, \mathbf{A}_{(d)}, \mathbf{w}, \sigma)$.
   (k) Let the user revocation token be **grt**[$d$] = $\mathbf{A}_{(d)}\cdot\mathbf{t}^{(d)}$.

Finally, the algorithm returns, **gpk** = $((\mathbf{A}, \mathbf{u}, \mathbf{w}), \mathbf{B})$, **gmsk** = $\mathbf{T}_\mathbf{B}$, **gsk** = $(\mathbf{gsk}[0], \mathbf{gsk}[1], \ldots, \mathbf{gsk}[N-1])$, **grt** = $(\mathbf{grt}[0], \mathbf{grt}[1], \ldots, \mathbf{grt}[N-1])$.

**Signing:** We employ the $\mathcal{OTS}$ scheme to secure the generating signatures. Then, we employ the underlying argument system to prove that the user is valid. Sign(**gpk**, **gsk**[$d$], **grt**[$d$], $M$) takes **gpk**,

the signer's secret signing key **gsk**[$d$], revocation token **grt**[$d$] as inputs and produces $\Sigma$ on a message $M$ as follows.

Let $\mathcal{H}_1$: $\{0,1\}^* \rightarrow \mathbb{Z}_q^{n\times\ell}$, $\mathcal{H}_2$: $\{0,1\}^* \rightarrow \{1,2,3\}^t$, and $\mathcal{G}$: $\{0,1\}^* \rightarrow \mathbb{Z}_q^{n\times m}$ be hash functions, modeled as a random oracle.

1. Run $\mathsf{OGen}(1^n)$ and get (**ovk, osk**).
2. Encrypt the signer's index $d$ as below.
   (a) Let $\mathbf{G} = \mathcal{H}_1(\mathbf{ovk})$.
   (b) Sample $\mathbf{s} \leftarrow \chi^n$, $\mathbf{e}_1 \leftarrow \chi^m$ and $\mathbf{e}_2 \leftarrow \chi^\ell$.
   (c) Compute the ciphertext $(\mathbf{c}_1, \mathbf{c}_2)$ pair which encrypts the index $d$ ($\mathbf{c}_1 = \mathbf{B}^T\mathbf{s} + \mathbf{e}_1$, $\mathbf{c}_2 = \mathbf{G}^T\mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor d$).
3. Sample $\rho \xleftarrow{\$} \{0,1\}^n$, and get $\mathbf{V} = \mathcal{G}(\mathbf{A}, \mathbf{u}, \mathbf{w}, \mathbf{B}, M, \rho) \in \mathbb{Z}_q^{m\times n}$.
4. Compute $\mathbf{v} = \mathbf{V} \cdot (\mathbf{A}_{(d)} \cdot \mathbf{t}^{(d)}) + \mathbf{e}_1 \mod q$ ($\|\mathbf{e}_1\|_\infty \leq \beta$ with overwhelming probability and $\mathbf{A}_{(d)} \cdot \mathbf{t}^{(d)} = \mathbf{grt}[d]$).
5. Generate the parameters for the interactive protocol to show the index $d$ is encrypted correctly as follows.

$$\mathbf{P} = \left( \begin{array}{c|c} \mathbf{B}^T & \\ \hline \mathbf{G}^T & \mathbf{I}_{m+\ell} \end{array} \right) \in \mathbb{Z}_q^{k_1\times k_2}; \mathbf{c} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} \in \mathbb{Z}^{k_1}; \mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{k_2}. \quad (9)$$

6. Repeat the underlying interactive protocol given in Section 4, $t = \omega(\log n)$ times with the public parameters (**A**, **u**, **w**, **B**, **V**, **v**, **P**, **c**) and prover's witness (**x**, **t**, $\mathbf{e}_1$, **e**) to achieve the soundness error negligible. Then make it non-interactive by employing the Fiat–Shamir heuristic as a triple,

   $\Pi = (\{CMT^{(k)}\}_{k=1}^t, CH, \{RSP^{(k)}\}_{k=1}^t)$, where

   $\mathrm{CH} = (\{Ch^{(k)}\}_{k=1}^t) = \mathcal{H}_2(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$.
7. Get $\mathcal{OTS}; sig = \mathsf{OSig}(\mathbf{osk}, (\mathbf{c}_1, \mathbf{c}_2, \Pi))$.
8. Return the signature $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$.

**Verification:** On input **gpk**, $RL = \{\{\mathbf{u}_i\}_i\} \subset \mathbb{Z}_q^n$, $M$, and $\Sigma$, the algorithm Verify confirms $\Sigma$ is valid on $M$ and signer is a valid member by executing the following steps.

1. Parse $\Sigma$ as $(\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$.
2. Get $\mathbf{V} = \mathcal{G}(\mathbf{A}, \mathbf{u}, \mathbf{w}, \mathbf{B}, M, \rho) \in \mathbb{Z}_q^{m\times n}$.
3. If $\mathsf{OVer}(\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig) = 0$ then output invalid and abort.
4. Parse $\Pi$ as $(\{CMT^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{RSP^{(k)}\}_{k=1}^t)$.
5. If $(Ch^{(1)}, \ldots, Ch^{(t)}) \neq \mathcal{H}_2(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$, then return invalid else continue.
6. For $k = 1$ to $t$ execute the verification steps of the commitment scheme in Section 4 with public parameters (**A**, **u**, **w**, **B**, **V**, **v**, **P**, **c**) to validate $RSP^{(k)}$ with respect to $CMT^{(k)}$ and $Ch^{(k)}$. If any of the conditions does not hold then return invalid and abort.
7. For each $\mathbf{u}_i \in RL$ compute $\mathbf{e}_i' = \mathbf{v} - \mathbf{V} \cdot \mathbf{u}_i \mod q$ to verify whether there exists an index $i$ satisfying $\|\mathbf{e}_i'\|_\infty \leq \beta$. If so output invalid.
8. Return valid.

**Open:** To identify the signer of the given message $M$ signature $\Sigma$ pair $\mathsf{Open}(\mathbf{gmsk}, M, \Sigma)$ functions as follows. Here, $\mathbf{gmsk} = \mathbf{T_B}$ is the group manager's secret key and $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig)$.

1. Let $\mathbf{G} = [\mathbf{g}_1 | \ldots | \mathbf{g}_\ell] = \mathcal{H}_1(\mathbf{ovk})$.

2. Then, for $i \in [\ell]$, sample $\mathbf{y}_i \leftarrow \mathsf{SamplePre}(\mathbf{T_B}, \mathbf{B}, \mathbf{g}_i, \sigma)$.
3. Let $\mathbf{Y} = [\mathbf{y}_1 | \ldots | \mathbf{y}_\ell] \in \mathbb{Z}^{m \times \ell}$.
4. Compute $d' = (d'_1, \ldots, d'_\ell) = \mathbf{c}_2 - \mathbf{Y}^T \mathbf{c}_1 \in \mathbb{Z}_q^\ell$.
5. For each $i \in [\ell]$ check whether $d'_i$ is closer to 0 than $\lfloor q/2 \rfloor \bmod q$. If so $d_i = 0$ else 1.
6. Return index $d = (d_1, \ldots, d_\ell) \in \{0,1\}^\ell$.

## 6. Correctness and Security Proof of the Scheme

### *6.1. Correctness*

We use the techniques in [14] and adapt the scheme provided in [10]. Even though we changed the revocation token generation with regards to the work in [10], there is no impact to the correctness of the scheme from new revocation token generation, as we check the signer's authenticity with *RL* separately and we prove the honesty of the token independently in zero-knowledge proof.

For all $n$, $N$, all (**gpk**, **gmsk**, **gsk**, **grt**) returned by $\mathsf{KeyGen}(n, N)$, all $d \in \{0, 1, \ldots, N-1\}$, and all $M \in \{0,1\}^*$, $\mathsf{Verify}$(**gpk**, *RL*, *M*, $\mathsf{Sign}$(**gpk**, **gsk**[*d*], **grt**[*d*], *M*)) = valid; **grt**$[d] \notin RL$ and $\mathsf{Open}$(**gmsk**, *M*, $\mathsf{Sign}$(**gpk**, **gsk**[*d*], **grt**[*d*], *M*)) = *d*.

We use the proof of correctness given in [10]. We prove the correctness of Open additionally.

**Lemma 1** ([10], Lemma 4). *Let* $\beta = \mathsf{poly}(n), q \geq (4\beta+1)^2$ *and* $m \geq 3n$. *Over the randomness of* $\mathbf{V} \in \mathbb{Z}_q^{m \times n}$,

$$Pr[\exists \text{ non-zero } \boldsymbol{s} \in \mathbb{Z}_q^n : \|\boldsymbol{V} \cdot \boldsymbol{s}\|_\infty \leq 2\beta] \leq \mathit{negl}(n).$$

*proof. Fix a non-zero vector* $\boldsymbol{s} \in \mathbb{Z}_q^n$. *Then the vector* $\boldsymbol{V} \cdot \boldsymbol{s}$ *is uniformly distributed over* $\mathbb{Z}_q^m$. *It then follows that* $Pr[\|\boldsymbol{V} \cdot \boldsymbol{s}\|_\infty \leq 2\beta] \leq \frac{(4\beta+1)^m}{q^m}$. *Applying a union-bound get*

$$Pr[\exists \text{ non-zero } \boldsymbol{s} \in \mathbb{Z}_q^n : \|\boldsymbol{V} \cdot \boldsymbol{s}\|_\infty \leq 2\beta] \leq \frac{q^n(4\beta+1)^m}{q^m} \leq \frac{1}{(4\beta+1)^{m-2n}} \leq (4\beta+1)^{-n} = \mathit{negl}(n).$$

An honest signer is able to obtain a valid witness $(\mathbf{x}, \mathbf{t}, \mathbf{e}_1, \mathbf{e})$ for the underlying argument system with overwhelming probability. Moreover, the verification algorithm Verify will not return Invalid for the signature check after Step 6, because Step 6 validates the signature using the underlying zero-knowledge interactive protocol. In Step 7 of the verification algorithm Verify, the vector $\mathbf{e}'_i$ for every $i$ can be delivered as

$$\mathbf{e}'_i = \mathbf{v} - \mathbf{V} \cdot \mathbf{u}_i = \mathbf{V} \cdot \mathbf{grt}[d] + \mathbf{e}_1 - \mathbf{V} \cdot \mathbf{u}_i = \mathbf{V} \cdot (\mathbf{grt}[d] - \mathbf{u}_i) + \mathbf{e}_1 \mod q.$$

If the verification algorithm Verify outputs Valid, that is $\|\mathbf{e}'_i\|_\infty \leq \beta$, for all $i$. This means $\mathbf{grt}[d] \notin RL$. If there exists an index $i$, where $\mathbf{grt}[d] = \mathbf{u}_i$, then $\mathbf{e}'_i = \mathbf{e}_1$. Then, the signature should not pass the Step 7 of the verification process because $\|\mathbf{e}'_i\|_\infty = \|\mathbf{e}_1\|_\infty \leq \beta$.

Suppose there is a situation $\mathbf{grt}[d] \notin RL$, i.e., for every $i$, the vector $\mathbf{s}_i := \mathbf{grt}[d] - \mathbf{u}_i \mod q$ is non-zero. According to Lemma 1, $\|\mathbf{V} \cdot \mathbf{s}_i\|_\infty > 2\beta$ with overwhelming probability. At the same time, $\|\mathbf{V} \cdot \mathbf{s}_i\|_\infty \leq \|\mathbf{e}'_i\|_\infty + \|\mathbf{e}_1\|_\infty \leq \|\mathbf{e}'_i\|_\infty + \beta$. Thus, $\|\mathbf{e}'_i\|_\infty > 2\beta - \beta = \beta$.

In our scheme, the revocation token $\mathbf{grt}[d] = (\mathbf{A}_{(d)}\mathbf{t}^{(d)})$ of a member $d$ is generated by using the public parameters $\mathbf{A}$ and $\mathbf{w}$. Thus, the underlying zero-knowledge protocol of the verification algorithm checks whether the user token satisfies $\mathbf{A}_{(d)}\mathbf{t}^{(d)} = \mathbf{w} \mod q$ in zero-knowledge. Thus, using the public parameters without revealing the $\mathbf{A}_{(d)}$ or $\mathbf{t}^{(d)}$ the signer should proof that his token is not being faked. On the other hand, because of this condition dishonest signers cannot generate a valid signature with a fake token that passes the signature verification. This confirms that only correct and honestly generated signatures can pass the signature verification.

Moreover, if the index of the signer is correctly encrypted in the ciphertext **c** at the time of signing, then the tracing algorithm Open returns the index of the signer correctly. Encryption of the index is guaranteed in the signing stage because no member can pass the underlying interactive protocol without correct encryption of the index via a LWE function. In addition to that, Verify returns Invalid if the ciphertext **c** is not correct encryption of the index because it cannot pass the underlying interactive protocol's checking without a correct encryption. Thus, this proves the correctness of the encryption of the index and that the tracing algorithm outputs the index of the correct signer.

### *6.2. Almost-Full Anonymity*

**Theorem 1.** *The proposed VLR group signature Section 5.1 is almost-full anonymous in the random oracle model under the* $LWE_{n,q,\chi}$ *assumption.*

We demonstrate the anonymity of our scheme using eight indistinguishable games, where $Adv_A(G_0) = \epsilon$ and $Adv_A(G_7) = 0$.

**Game** $\mathbf{G}_0$: This is the naive anonymity experiment. Here, we believe that the adversary $A$ has advantage $\epsilon$. At the beginning, the challenger $C$ produces keys **gpk**, **gmsk**, $\mathbf{gsk}[d]_{d\in\{0,1\}^\ell}$, and $\mathbf{grt}[d]_{d\in\{0,1\}^\ell}$ using KeyGen($n$, $N$). Then, he/she hand overs **gpk** and **gsk** to $A$. The adversary $A$ is allowed to query revocation tokens and opening of any signature. When $A$ asks for a token of a user $d$ the challenger $C$ outputs **grt**[$d$]. When $A$ queries for opening of any signature then $C$ answers with $\mathsf{Open}(\mathbf{gmsk}, M, \Sigma)$ using **gmsk** = $\mathbf{T_B}$. In the challenge phase, $A$ sends a message $M$ and two indices $i_0, i_1 \in \{0,1\}^\ell$, such that the adversary $A$ did not make a revocation query for users $i_0, i_1 \in \{0,1\}^\ell$. Then, $C$ sends back $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$, which is generated using $\mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i_b], \mathbf{grt}[i_b], M)$. The adversary $A$ still can query for opening oracle except for challenged indices and he is not allowed for revocation queries with $i_0, i_1$. The adversary's challenge is to guess the index, that is employed to produce $\Sigma^*$. Finally, $A$ returns $b' \in \{0,1\}$. If $b' = b$ then $A$ wins.

**Game** $\mathbf{G}_1$: In this game, a minor modification is done compared to $G_0$. The $\mathcal{OTS}$ key pair $(\mathbf{ovk}^*, \mathbf{osk}^*)$, which is created at the signature generation in the real game, is produced at the initial stage of the game. Thus, at the query phase, if $A$ queries for opening oracle with $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$, where $\mathbf{ovk} = \mathbf{ovk}^*$ then the challenger $C$ sends a random bit and terminates the game. As $\mathbf{ovk}^*$ is created at the start, it does not have any relation to the adversary's queries. Accordingly, the probability of $\mathbf{ovk} = \mathbf{ovk}^*$ is insignificant. Besides, after the challenge signature $\Sigma^* = (ovk^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ is sent, if the adversary queries a valid signature $\Sigma = (ovk, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$ with $\mathbf{ovk} = \mathbf{ovk}^*$ then $sig$ is a forged one. Therefore, the challenger terminating the game is irrelevant. Without losing the generality we believe that $A$ does not ask for opening of a valid $\Sigma$ with $\mathbf{ovk} = \mathbf{ovk}^*$.

**Game** $\mathbf{G}_2$: Here, we replace the encrypting matrices **B** and **G** with randomly obtained $\mathbf{B}^*$ and $\mathbf{G}^*$, and we program the random oracle $\mathcal{H}_1$ according to **B** and **G**. In real anonymity game, **B** is obtained from GenTrap and **G** is generated at the signature generation. In this game, we get uniformly random $\mathbf{B}^* \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{G}^* \in \mathbb{Z}_q^{n\times \ell}$. The challenger samples $\mathbf{Y} \leftarrow (D_{z^m,\sigma})^\ell$ compute $\mathbf{G} = \mathbf{B}^*\mathbf{Y} \in \mathbb{Z}_q^{n\times\ell}$ and program $\mathcal{H}_1(\mathbf{ovk}^*) = \mathbf{G}$ to respond the opening oracles querying with $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$. This **G** is utilized to respond the opening and (**ovk**,**Y**,**G**) is saved to reuse when $A$ repeats the same queries for $\mathcal{H}_1(\mathbf{ovk})$. In the challenge phase, program $\mathcal{H}_1(\mathbf{ovk})^* = \mathbf{G}^*$ and compute $(\mathbf{c}_1^*, \mathbf{c}_2^*)$ to make $\Sigma^*$. As the distribution of **G** is statistically close to uniform over $\mathbb{Z}_q^{n\times\ell}$ [29] and the distributions of $\mathbf{G}^*, \mathbf{B}^*$ are statistically close to the real game [29], the games $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable.

**Game** $\mathbf{G}_3$: Without producing the legitimate non-interactive proof $\Pi$, in this game, $C$ simulates $\Pi$ as discussed in Section 4. For each $k \in [t]$, take a forged challenge $\overline{Ch}^{(k)}$ and execute the interactive protocol. Then, program the random oracle $\mathcal{H}_1$ respectively. The challenging signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ is statistically close to the signature in the early games because the argument system is statistically zero-knowledge. Therefore, $G_3$ is indistinguishable from $G_2$.

**Game** $\mathbf{G}_4$: The challenger replaces the naive revocation token used to generate the challenged signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$, where $\mathbf{v} = \mathbf{V} \cdot \mathbf{grt}[i_b] + \mathbf{e}_1 \mod q$, with a vector $\mathbf{t}$ sampled uniformly. We compute $\mathbf{v} = \mathbf{V} \cdot \mathbf{t} + \mathbf{e}_1 \mod q$, where $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_q^n$. $\mathbf{V}$ is uniformly random over $\mathbb{Z}_q^{m \times n}$, $\mathbf{e}_1$ sampled from the error distribution $\chi$, and we replace only $\mathbf{grt}[i_b]$ by $\mathbf{t}$. As the rest of the game is the same as the previous game $G_3$, this game is statistically indistinguishable from $G_3$.

**Game** $\mathbf{G}_5$: In this game, we obtain $\mathbf{v}$ uniformly. Thus, we make the revocation token totally independent of the challenging bit $b$. We sample $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ and set $\mathbf{v} = \mathbf{y}$. In the above game, the pair $(\mathbf{V}, \mathbf{v})$ is a proper $LWE_{n,q,\chi}$ instance, and in this game we replace $\mathbf{v}$ with truly uniformly sampled $\mathbf{y}$. As the $LWE_{n,q,\chi}$ problem is hard (Section 2), the games $G_4$ and $G_5$ are indistinguishable. Suppose there is an algorithm B for solving the $LWE_{n,q,\chi}$ problem. Then, B can interact with $A$ by answering the queries that $A$ makes. When $A$ queries for the revocation token of any group member, B can simply answer with a value chosen uniformly random such as $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ instead of providing $\mathbf{grt}$. The rest of the game is the same as the original poof given in the previous game. If adversary $A$ can distinguish whether the revocation token is generated or chosen randomly, the algorithm B succeeds. However, this contradicts the hardness of the $LWE_{n,q,\chi}$ problem.

**Game** $\mathbf{G}_6$: In this game, we modify the creation of ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*)$ uniformly. Let $\mathbf{c}_1^* = \mathbf{x}_1$ and $\mathbf{c}_2^* = \mathbf{x}_2 + \lfloor q/2 \rfloor d_b$, where $\mathbf{x}_1 \in \mathbb{Z}^m$ and $\mathbf{x}_2 \in \mathbb{Z}^\ell$ are uniformly random and $d_b$ is the index of the challenger's bit. As the rest of the game is same as $G_5$ and the $LWE_{n,q,\chi}$ problem is hard to solve, the games $G_5$ and $G_6$ are indistinguishable. Indeed, if $A$ succeed on distinguishing two games, then $A$ can also solve the LWE problem. That means, he/she can distinguish $(\mathbf{B}^*, (\mathbf{B}^*)^T\mathbf{s} + \mathbf{e}_1)$ from $(\mathbf{B}^*, \mathbf{z}_1)$ and $(\mathbf{G}^*, (\mathbf{G}^*)^T\mathbf{s} + \mathbf{e}_2)$ from $(\mathbf{G}^*, \mathbf{z}_2)$, which conflicts with the $LWE_{n,q,\chi}$ assumption.

**Game** $\mathbf{G}_7$: The challenger makes the challenging signature $\Sigma^*$ totally independent of the bit $b$. Let $\mathbf{c}_1^* = \mathbf{x}_1'$ and $\mathbf{c}_2^* = \mathbf{x}_2'$, where $\mathbf{x}_1' \in \mathbb{Z}_q^m$ and $\mathbf{x}_2' \in \mathbb{Z}_q^\ell$ are uniformly random. The games $G_6$ and $G_7$ are statistically indistinguishable. As this game $G_7$ is independent from the challenger's bit $b$, the advantage of the adversary winning the game $Adv_A$ is 0.

Therefore, the above executed games prove that advantage of the adversary on almost-full anonymity of the scheme is negligible.

This concludes the proof of anonymity.

### *6.3. Traceability*

In the random oracle model, we say that our VLR group signature scheme is traceable if the $\mathbf{SIS}^\infty_{n,(\ell+1)m,q,2\beta}$ problem is hard.

**Lemma 2** ([29]). *For any $m$, $\beta = poly(n)$, and for any $q \geq \beta.\omega(\sqrt{n \log n})$, solving a random instance of the $\boldsymbol{SIS}^2_{n,m,q,\beta}$ or $\boldsymbol{ISIS}^2_{n,m,q,\beta}$ problem with non-negligible probability is at least as hard as approximating the $\boldsymbol{SIVP}^2_\gamma$ problem on any lattice of dimension $n$ to within certain $\gamma = \beta \cdot \tilde{\mathcal{O}}(\sqrt{n})$ factors.*

**Theorem 2** ([10]). *If there is a traceability adversary $A$ with success probability $\epsilon$ and running time $\mathrm{T}$, then there is an algorithm $\mathcal{F}$ that solves the $\boldsymbol{SIS}^\infty_{n,(\ell+1).m,q,2\beta}$ problem with success probability $\epsilon' > (1 - (7/9)^t) \cdot 1/2N$, and running time $\mathrm{T}' = 32 \cdot \mathrm{T}.q_{\mathcal{H}}/(\epsilon - 3^{-t}) + poly(n, N)$, where $q_{\mathcal{H}}$ is the number of queries to the random oracle $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$.*

Based on Lemma 2 and Theorem 2, we prove that the proposed scheme is traceable in the random oracle.

Suppose there is an adversary $A$ who can break the computational binding property of the commitment scheme COM with non-negligible probability. Therefore, $A$ can find answers for the $\mathbf{SIS}^\infty_{n,(\ell+1)m,q,2\beta}$ problem. Thus, without losing the generality, we believe that COM is computationally binding.

Let forger $\mathcal{F}$ be a PPT algorithm that solves the $\mathbf{SIS}^{\infty}_{n,(\ell+1)m,q,2\beta}$ problem with non-negligible probability.

The forgery $\mathcal{F}$ is given the verification key $(\mathbf{A}, \mathbf{u}, \mathbf{w})$. $\mathcal{F}$ then produces a key pair $(\mathbf{B}, \mathbf{T_B})$ and interacts with the adversary $A$ by sending $\mathbf{gpk} = ((\mathbf{A}, \mathbf{u}), \mathbf{B})$ and responding to the $A$'s queries as below.

- **Signatures queries**: If the adversary $A$ asks a signature of user $d$ on a random message $M$, then $\mathcal{F}$ outputs $\Sigma = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[d], \mathbf{grt}[d], M) = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi', sig, \mathbf{v}, \rho)$, where $\Pi'$ is simulated without using the legitimate secret key and others are generated faithfully. The zero-knowledge property of the given underlying interactive protocol guarantees that $\Sigma$ is indistinguishable from the legitimate group signature.
- **Corruption queries**: The corruption set $CU$ is initially a empty set. If the adversary $A$ asks the secret signing key of any user $d$, then $\mathcal{F}$ adds $d$ to $CU$ and returns $\mathbf{gsk}[d]$.
- Queries to the random oracles $\mathcal{H}_1$ and $\mathcal{H}_2$ are handled by consistently returning uniformly random values in $\{1, 2, 3\}^t$. For each $k \leq q_{\mathcal{H}}$, $r_k$ denotes the answer to the $k$-th query.

Finally, $A$ sends a message $M^*$, revocation data $RL^*$, and a non-trivial forged signature $\Sigma^*$, which fulfills the requirements of the traceability game, where $\Sigma^* = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), M, (\{CMT^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{RSP^{(k)}\}_{k=1}^t), sig, \mathbf{v}, \rho)$, such that $\mathsf{Verify}(\mathbf{gpk}, M^*, RL^*, \Sigma^*) = \mathsf{Valid}$ and Open fails or outputs an user index outside of the coalition $CU \setminus RL^*$

Now let us show how $\mathcal{F}$ exploits the forgery.

We assume that $A$ always queries $\mathcal{H}_2$ on input $(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$ before $\mathcal{H}_1$. Thus, with probability at least $\epsilon - 3^{-t}$, there exists certain $k^* \leq q_{\mathcal{H}}$ such that the $k^*$-th oracle queries involves the tuple $(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$. Then, for any fixed $k^*$, execute $A$ many times and input as in the original run. For each repeated execution, $A$ gives the same results for the first $k^*$-1 queries as in the initial run and from the $k^*$-th query onward he/she outputs fresh random values. As stated in the forking lemma [[37], Lemma 7], with probability larger than $1/2$, algorithm $\mathcal{F}$ can get a 3-fork involving tuple $(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$ after less than $32 \cdot q\mathcal{H}/(\epsilon - 3^{-t})$ executions of $A$. Let the outputs of $\mathcal{F}$ with respect to the 3-fork branches be

$$r^{(1)}_{\kappa^*} = (Ch^{(1)}_1, \ldots, Ch^{(1)}_t); r^{(2)}_{\kappa^*} = (Ch^{(2)}_1, \ldots, Ch^{(2)}_t); r^{(3)}_{\kappa^*} = (Ch^{(3)}_1, \ldots, Ch^{(3)}_t).$$

A simple computation shows that
$Pr[\exists j \in \{1, \ldots, t\}] : \{Ch^{(1)}_i, Ch^{(2)}_i, Ch^{(3)}_i\} = \{1, 2, 3\}1 - (7/9)^t$.

Under the condition of the existence of such index $i$, one parses the three forgeries related to the fork branches to get $(RSP^{(1)}_i, RSP^{(2)}_i, RSP^{(3)}_i)$.

Then, by employing the knowledge extractor of the underlying argument system, we can obtain vectors $(\mathbf{y}, \mathbf{e}^*_1, \mathbf{t}^*, \mathbf{e}^*)$. We can get $(\mathbf{s}^*, \mathbf{e}^*_1, \mathbf{e}^*_2)$ from $\mathbf{e}^*$, which satisfy the following.

1. $\mathbf{y} = (\mathbf{y}_0 \| \mathbf{y}^0_1 \| \mathbf{y}^1_1 \| \ldots \| \mathbf{y}^0_\ell \| \mathbf{y}^1_\ell) \in \mathsf{Secret}_\beta(d)$ 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 $\mathbf{V} \cdot (\mathbf{A}_{(d)} \cdot \mathbf{t}) + \mathbf{f}' = \mathbf{v} \mod q$.
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. $\mathbf{A}_{(d)} \cdot \mathbf{t} = \mathbf{w} \mod q$.

We can check that $(\mathbf{c}_1, \mathbf{c}_2)$ is a correct encryption of $d^*$, the tracing algorithm $\mathsf{Open}(\mathbf{T_B}, M^*, \Sigma^*)$ returns $d^*$, $\mathsf{Verify}(\mathbf{gpk}, \Sigma, M,^* \mathbf{grt}[j^*]) = \mathsf{Invalid}$ and $\mathsf{Verify}(\mathbf{gpk}, \Sigma, M,^* RL^*) = \mathsf{Valid}$.

It then follows that $\mathbf{grt}[j^*] \notin RL$ and $j^* \notin CU$. As a result, $(\mathbf{y}, d^*)$ is a valid forgery. Furthermore, the analysis of the forgery signature shows that if $A$ has non-negligible success probability returns in polynomial time, then so does $\mathcal{F}$.

This satisfies the proof of traceability.

## 7. Conclusion and Open Problems

This paper presented a group signature scheme from lattices that provides member revocation facility using VLR, which is the most efficient revocation approach up to now, while being almost-full anonymous. Moreover, the scheme provides an explicit tracing algorithm Open which can be used to trace a signer in a large group efficiently. However, delivering an efficient VLR group signature scheme with full security is a challenging task which is not yet solved.

**Author Contributions:** Conceptualization, M.N.S.P.; writing—original draft preparation, M.N.S.P.; writing—review and editing, M.N.S.P. and T.K.; supervision, T.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by JSPS Grants-in-Aid for Scientic Research Numbers 16H01705 and 17H01695.

**Acknowledgments:** Some part of this paper are result of research by Advanced Telecommunications Research Institute International (ATR), Japan.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Analysis of the Underlying Zero-Knowledge Protocol

Let **COM** be a statistically hiding and computationally binding string commitment scheme. The interactive protocol is a zero-knowledge argument of knowledge with perfect completeness and soundness error 2/3 with $(\mathcal{O}(\ell m)\log\beta + \mathcal{O}(k_2)\log b)\log q$ communication cost. Thus, it satisfies the following.

- There exists an efficient simulator that, on input $(\mathbf{A}, \mathbf{u}, \mathbf{w}, \mathbf{B}, \mathbf{V}, \mathbf{v}, \mathbf{P}, \mathbf{c})$, outputs an accepted transcript which is statistically close to that produced by the real prover.
- There exists an efficient knowledge extractor that, on input a commitment $CMT$ and three valid responses $(RSP^{(1)}, RSP^{(2)}, RSP^{(3)})$ corresponding to all three possible values of the challenging Ch, outputs vectors $(\mathbf{y}, \mathbf{f}', \mathbf{r}', \mathbf{e}')$ such that

  1. $\mathbf{y} = (\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1) \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$, and $\mathbf{A} \cdot \mathbf{y} = \mathbf{u} \mod q$.
  2. $\|\mathbf{f}'\|_\infty \leq \beta$ and $\mathbf{V} \cdot (\mathbf{A}_{(d)} \cdot \mathbf{t}) + \mathbf{f}' = \mathbf{v} \mod q$.
  3. $\|\mathbf{e}'\|_\infty \leq b$ and $\mathbf{P}\mathbf{e}' + (0^{k_1-\ell} \| \lfloor q/2 \rfloor d) = \mathbf{c} \mod q$.
  4. $\mathbf{A}_{(d)} \cdot \mathbf{t} = \mathbf{w} \mod q$.

### *Appendix A.1. Completeness and Soundness*

An honest prover, with a valid witness $(\mathbf{x}, \mathbf{f}, \mathbf{t}, \mathbf{e})$ for some $d \in \{0,1\}^\ell$, can always obtain $\mathbf{z}_1, \ldots, \mathbf{z}_p \in \mathsf{Secret}_\beta(d), \mathbf{f}_1, \ldots, \mathbf{f}_p \in \mathsf{B}_{3m}, \mathbf{t}_1, \ldots, \mathbf{t}_p \in \mathsf{B}_{3m}$, and $\mathbf{e}_1, \ldots, \mathbf{e}_{\bar{p}} \in \mathsf{B}_{3k_2}$ via the Decomposition-Extension technique [10]. If he/she follows the protocol, he/she should always be accepted by the verifier. In this manner, the protocol has perfect completeness.

The protocol admits a soundness error 2/3, which is natural for typical Stern-like protocols. However, this error can be made negligible by repeating the protocol $t = \omega(\log n)$ times in parallel.

### *Appendix A.2. Communication Cost*

The KTX scheme [36] COM outputs an element of $\mathbb{Z}_q^n$. Therefore, the commitment CMT has bit size $3n \log q = \tilde{\mathcal{O}}(n)$. The response RSP is executed by $p$ permutations in $S$, $2p$ permutations in $S_{3m}$, $\bar{p}$ permutations in $S_{3k_2}$, one permutation in $2\ell$, $p$ vectors in $\mathbb{Z}_q^{(2\ell+1)3m}$, $2p$ vectors in $\mathbb{Z}_q^{3m}$, $\bar{p}$ vectors in $\mathbb{Z}_q^{3k_2}$, and one vector in $\mathbb{Z}_q^{2\ell}$.

In this manner, the bit size of RSP is bounded by $(\mathcal{O}(\ell m)p + \mathcal{O}(k_2)\bar{p})\log q$, where $p = \lfloor \log \beta \rfloor + 1$ and $p = \lfloor \log b \rfloor + 1$. Thus, the overall communication cost of the protocol is bounded by $(\mathcal{O}(\ell m)\log\beta + \mathcal{O}(k_2)\log b)\log q$.

*Appendix A.3. Zero-Knowledge Property*

If **COM** is statistically hiding, we can prove that the interactive protocol is statistical zero-knowledge argument.

First, construct a PPT simulator *SIM* interacting with a verifier $\hat{\mathcal{V}}$ such that, by giving only the public inputs, *SIM* outputs with probability close to 2/3 a simulated transcript that is statistically close to the outputs of an honest prover in the real interaction. From the public input (**A**, **u**, **w**, **B**, **V**, **v**, **P**, **c**) given by the protocol, both *SIM* and $\hat{\mathcal{V}}$ acquire matrices, $\mathbf{A}^*$, $\mathbf{V}^*$, $\mathbf{I}^*$, $\mathbf{P}^*$, and **Q**. Then, *SIM* starts simulation by selecting a random $\overline{Ch} \in \{1,2,3\}$. This is a prediction of the challenge value that $\hat{\mathcal{V}}$ will not choose.

**Case** $\overline{Ch} = 1$ : *SIM* computes the vectors $\mathbf{z}'_1, \ldots, \mathbf{z}'_p \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}'_j) = \mathbf{u} \mod q$, $\mathbf{t}'_1, \ldots, \mathbf{t}'_p \in \mathbb{Z}_q^{3m}$ and $\mathbf{f}'_1, \ldots, \mathbf{f}'_p \in \mathbb{Z}_q^{3m}$ such that $\mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{t}'_j) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j) = \mathbf{v} \mod q$ and $\mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{t}') = \mathbf{w} \mod q$, and $\mathbf{e}'_1, \ldots, \mathbf{e}'_{\bar{p}} \in \mathbb{Z}_q^{3k}$ and $d' \in \mathbb{Z}_q^{2\ell}$, such that $\mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{e}'_j) + \mathbf{Q} \cdot d' = \mathbf{c} \mod q$ by using linear algebra.

Then, *SIM* samples objects given below as in Equation (1).

$$\begin{cases} c \xleftarrow{\$} \{0,1\}^{\ell}; \\ \pi_{z,1}, \ldots, \pi_{z,p} \xleftarrow{\$} S; \pi_{f,1}, \ldots, \pi_{f,p} \xleftarrow{\$} S_{3m}; \pi_{t,1}, \ldots, \pi_{t,p} \xleftarrow{\$} S_{3m}; \\ \quad \pi_{e,1}, \ldots, \pi_{e,\bar{p}} \xleftarrow{\$} S_{3k_2}; \tau \xleftarrow{\$} S_{2\ell}; \\ \mathbf{r}_{z,1}, \ldots, \mathbf{r}_{z,p} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)3m}; \mathbf{r}_{f,1}, \ldots, \mathbf{r}_{f,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \\ \quad \mathbf{r}_{t,1}, \ldots, \mathbf{r}_{t,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{r}_{e,1}, \ldots, \mathbf{r}_{e,\bar{p}} \xleftarrow{\$} \mathbb{Z}_q^{3k_2}; \mathbf{r}_d \xleftarrow{\$} \mathbb{Z}_q^{2\ell}. \end{cases} \tag{A1}$$

sends commitment $\mathbf{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\hat{\mathcal{V}}$, where

$$\begin{cases} \mathbf{c}'_1 = \mathsf{COM}(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{t,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{z,j}); \\ \qquad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{t,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{f,j}); \\ \qquad \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{t,j}); \\ \qquad \{\pi_{e,j}\}_{j=1}^{\bar{p}}, \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{r}_{e,j}) + \mathbf{Q}\mathbf{r}_d; \tau; \rho_1), \\ \mathbf{c}'_2 = \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{r}_{z,j}), \pi_{f,j}(\mathbf{r}_{f,j}), \pi_{t,j}(\mathbf{r}_{t,j})\}_{j=1}^{p}; \\ \qquad \{\pi_{e,j}(\mathbf{r}_{e,j})\}_{j=1}^{\bar{p}}; \tau(\mathbf{r}_d); \rho_2), \\ \mathbf{c}'_3 = \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{z}'_j + \mathbf{r}_{z,j}), \pi_{f,j}(\mathbf{f}'_j + \mathbf{r}_{f,j}), \pi_{t,j}(\mathbf{t}'_j + \mathbf{r}_{t,j})\}_{j=1}^{p}; \\ \qquad \{\pi_{e,j}(\mathbf{e}'_j + \mathbf{r}_{e,j})\}_{j=1}^{\bar{p}}; \tau(d' + \mathbf{r}_d); \rho_3). \end{cases} \tag{A2}$$

For a challenge *Ch* from $\hat{\mathcal{V}}$, *SIM* responds as follows.

- If *Ch* = 1: Output $\perp$ and abort.
- If *Ch* = 2: Send,

RSP = $(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{t,j}, \mathbf{z}'_j + \mathbf{r}_{z,j}, \mathbf{f}'_j + \mathbf{r}_{f,j}, \mathbf{t}'_j + \mathbf{r}_{t,j}\}_{j=1}^{p}$,
$\{\pi_{e,j}, \mathbf{e}'_j + \mathbf{r}_{e,j}\}_{j=1}^{\bar{p}}, d' + \mathbf{r}_d, \tau, \rho_1, \rho_3)$.

- If *Ch* = 3: Send,

RSP = $(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{t,j}, \mathbf{r}_{z,j}, \mathbf{r}_{f,j}, \mathbf{r}_{t,j}\}_{j=1}^{p}, \{\pi_{e,j}, \mathbf{r}_{e,j}\}_{j=1}^{\bar{p}}, \tau, \rho_1, \rho_2)$.

**Case** $\overline{Ch} = 2$ : *SIM* samples randomness $\rho_1, \rho_2, \rho_3$ for **COM** and

$$\begin{cases} \hat{d} \xleftarrow{\$} \{0,1\}^{\ell}, c \xleftarrow{\$} \{0,1\}^{\ell}; d' \xleftarrow{\$} \mathsf{B}_{2\ell}; \\ \mathbf{z}'_1, \ldots, \mathbf{z}'_p \xleftarrow{\$} \mathsf{SecretExt}(d); \mathbf{f}'_1, \ldots, \mathbf{f}'_p \xleftarrow{\$} \mathsf{B}_{3m}; \mathbf{t}'_1, \ldots, \mathbf{t}'_p \xleftarrow{\$} \mathsf{B}_{3m}; \\ \quad \mathbf{e}'_1, \ldots, \mathbf{e}'_{\bar{p}} \xleftarrow{\$} \mathsf{B}_{3k}; \\ \pi_{z,1}, \ldots, \pi_{z,p} \xleftarrow{\$} S; \pi_{f,1}, \ldots, \pi_{f,p} \xleftarrow{\$} S_{3m}; \pi_{t,1}, \ldots, \pi_{t,p} \xleftarrow{\$} S_{3m}; \\ \quad \pi_{e,1}, \ldots, \pi_{e,\bar{p}} \xleftarrow{\$} S_{3k}; \\ \mathbf{r}_{z,1}, \ldots, \mathbf{r}_{z,p} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)3m}; \mathbf{r}_{f,1}, \ldots, \mathbf{r}_{f,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{r}_{t,1}, \ldots, \mathbf{r}_{t,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \\ \quad \mathbf{r}_{e,1}, \ldots, \mathbf{r}_{e,\bar{p}} \xleftarrow{\$} \mathbb{Z}_q^{3k}; \mathbf{r}_d \xleftarrow{\$} \mathbb{Z}_q^{2\ell}, \tau \xleftarrow{\$} S_{2\ell}. \end{cases}$$

Next, *SIM* forms and sends commitment CMT as the same manner as in (A2).

For a challenge *Ch* from $\hat{\mathcal{V}}$, *SIM* responds as follows.

- If *Ch* = 1: $(\hat{d} \oplus c \ \{T_c \circ \pi_{z,j}(\mathbf{z}'_j), T_c \circ \pi_{z,j}(\mathbf{r}_{z,j}), \pi_{f,j}(\mathbf{f}'_j), \pi_{f,j}(\mathbf{r}_{f,j}),$
$\pi_{t,j}(\mathbf{t}'_j), \pi_{t,j}(\mathbf{r}_{t,j})\}_{j=1}^{p}, \{\pi_{e,j}(\mathbf{e}'_j), \pi_{e,j}(\mathbf{r}_{e,j})\}_{j=1}^{\bar{p}}, \tau(d'), \tau(\mathbf{r}_d))$.
- If *Ch* = 2: Output $\perp$ and abort.
- If *Ch* = 3: Send, RSP computed as in the case $(\overline{Ch} = 1, Ch = 3)$.

**Case** $\overline{Ch} = 3$ : *SIM* samples randomness as in $\overline{Ch} = 2$ and sends the commitment $\mathbf{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\hat{\mathcal{V}}$, where $\mathbf{c}'_2, \mathbf{c}'_3$ are computed as in (A2), and

$$\begin{aligned} \mathbf{c}'_1 = \mathsf{COM}\ (c, \{\pi_{z,j}, \pi_{e,j}, \pi_{t,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{z}'_j + \mathbf{r}_{z,j})) - \mathbf{u}; \\ \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{t}'_j + \mathbf{r}_{t,j})) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{f}'_j + \mathbf{r}_{f,j})) - \mathbf{v}; \\ \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{t}'_j + \mathbf{r}_{t,j})) - \mathbf{w}; \\ \{\pi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \sum_{j=1}^{\bar{p}} b_j (\mathbf{e}'_j + \mathbf{k}_{e,j}) + \mathbf{Q}(d' + \mathbf{k}_d) - \mathbf{c}; \tau; \rho_1). \end{aligned}$$

For a challenge *Ch* from $\hat{\mathcal{V}}$, *SIM* responds as follows.

- If *Ch* = 1: Send, RSP computed as in the case $(\overline{Ch} = 2, Ch = 1)$.
- If *Ch* = 2: Send, RSP computed as in the case $(\overline{Ch} = 1, Ch = 2)$.
- If *Ch* = 3: Output $\perp$ and abort.

As **COM** is statistically hiding, the distribution of the commitment CMT and the distribution of the challenge *Ch* from $\hat{\mathcal{V}}$ for every case considered above are statistically close to those in the real interaction. Therefore, the probability that the simulator outputs $\perp$ is negligibly close to 1/3. Thus, the simulator *SIM* can successfully imitate the honest prover with probability negligibly close to 2/3.

*Appendix A.4. Argument of Knowledge*

Here, we prove that if COM is computationally binding, then the given protocol is an argument of knowledge. For a given commitment CMT and three valid responses $RSP^{(1)}, RSP^{(2)}, RSP^{(3)}$ to all three possible values of the challenge *Ch*, a valid witness can be extracted.

$$\begin{cases} \mathbf{c}_1 & = \mathsf{COM}(d_2, \{\phi_{z,j}, \phi_{f,j}, \phi_{t,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{z,j}) - \mathbf{u}; \\ & \qquad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{t,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{f,j}) - \mathbf{v}; \\ & \qquad \mathbf{A}^* (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{t,j}) - \mathbf{w}; \\ & \qquad \{\phi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{s}_{e,j}) + \mathbf{Q}\mathbf{s}_d - \mathbf{c}; \hat{\tau}; \rho_1) \\ & = \mathsf{COM}(d_3, \{\psi_{z,j}, \psi_{f,j}, \psi_{t,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{z,j}); \\ & \qquad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{t,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{f,j}); \\ & \qquad \mathbf{A} * \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{t,j}) - \mathbf{w}; \\ & \qquad \{\psi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{h}_{e,j}) + \mathbf{Q}\mathbf{h}_d; \tilde{\tau}; \rho_1), \\ \mathbf{c}_2 & = \mathsf{COM}(\{\mathbf{b}_{z,j}, \mathbf{b}_{f,j}, \mathbf{b}_{t,j}\}_{j=1}^{p}, \{\mathbf{b}_{e,j}\}_{j=1}^{\bar{p}}, \mathbf{b}_d; \rho_2) \\ & = \mathsf{COM}(\{T_{d_3} \circ \psi_{z,j}(\mathbf{h}_{z,j}), \psi_{f,j}(\mathbf{h}_{f,j}), \psi_{t,j}(\mathbf{h}_{t,j})\}_{j=1}^{p}; \\ & \qquad \{\psi_{e,j}(\mathbf{h}_{e,j})\}_{j=1}^{\bar{p}}, \tilde{\tau}(\mathbf{h}_d); \rho_2), \\ \mathbf{c}_3 & = \mathsf{COM}(\{\mathbf{a}_{z,j} + \mathbf{b}_{z,j}, \mathbf{a}_{f,j} + \mathbf{b}_{f,j}, \mathbf{a}_{t,j} + \mathbf{b}_{t,j}\}_{j=1}^{p}; \\ & \qquad \{\mathbf{a}_{e,j} + \mathbf{b}_{e,j}\}_{j=1}^{\bar{p}}, \{\mathbf{a}_d + \mathbf{b}_d\}; \rho_3) \\ & = \mathsf{COM}(\{T_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}), \phi_{f,j}(\mathbf{s}_{f,j}), \phi_{t,j}(\mathbf{s}_{t,j})\}_{j=1}^{p}; \\ & \qquad \{\phi_{e,j}(\mathbf{s}_{e,j})\}_{j=1}^{\bar{p}}, \hat{\tau}(\mathbf{s}_d); \rho_3). \end{cases}$$

The computational binding property of **COM** implies that

$$\begin{cases} d_2 = d_3; \\ \mathbf{a}_d \in \mathsf{B}_{2\ell}; \hat{\tau} = \tilde{\tau}; \mathbf{b}_d = \tilde{\tau}(\mathbf{h}_d); \mathbf{a}_d + \mathbf{b}_d = \hat{\tau}(\mathbf{s}_d); \\ \forall j \in [p] : \phi_{z,j} = \psi_{z,j}; \mathbf{b}_{z,j} = T_{d_2} \circ \phi_{z,j}(\mathbf{h}_{z,j}) \textit{ and} \\ \qquad \mathbf{a}_{z,j} + \mathbf{b}_{z,j} = T_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}); \\ \forall j \in [p] : \phi_{f,j} = \psi_{f,j}; \mathbf{b}_{f,j} = \phi_{f,j}(\mathbf{h}_{f,j}) \textit{ and } \mathbf{a}_{f,j} + \mathbf{b}_{f,j} = \phi_{f,j}(\mathbf{s}_{f,j}); \\ \forall j \in [p] : \phi_{t,j} = \psi_{t,j}; \mathbf{b}_{t,j} = \phi_{t,j}(\mathbf{h}_{t,j}) \textit{ and } \mathbf{a}_{t,j} + \mathbf{b}_{t,j} = \phi_{t,j}(\mathbf{s}_{t,j}); \\ \forall j \in [\bar{p}] : \phi_{e,j} = \psi_{e,j}; \mathbf{b}_{e,j} = \phi_{e,j}(\mathbf{h}_{e,j}) \textit{ and } \mathbf{a}_{e,j} + \mathbf{b}_{e,j} = \phi_{e,j}(\mathbf{s}_{e,j}); \\ \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{z,j} - \mathbf{h}_{z,j})) = \mathbf{u} \mod q; \\ \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{t,j} - \mathbf{h}_{t,j})) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{f,j} - \mathbf{h}_{f,j})) = \mathbf{v} \mod q; \\ \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{t,j} - \mathbf{h}_{t,j})) = \mathbf{w} \mod q; \\ \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \mathbf{s}_{e,j}) + \mathbf{Q}\mathbf{s}_d - \mathbf{c} = \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \mathbf{h}_{e,j}) + \mathbf{Q}\mathbf{h}_d \mod q. \end{cases}$$

For each $j \in [p]$, let $\mathbf{y}'_j = (\mathbf{s}_{z,j} - \mathbf{h}_{z,j})$. Then, $T_{d_2} \circ \phi_{z,j}(\mathbf{y}'_j) = T_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}) - T_{d_2} \circ \phi_{z,j}(\mathbf{h}_{z,j}) = \mathbf{a}_{z,j} \in \mathsf{SecretExt}(d_1)$. Thus, $\phi_{z,j}(\mathbf{y}'_j) \in \mathsf{SecretExt}(d_1 \oplus d_2)$. Let $\bar{d} = d_1 \oplus d_2$, then for all $j \in [p]$, $\mathbf{y}'_j \in \mathsf{SecretExt}(\bar{d})$, as the permutation $\phi_{z,j} \in S$ preserves the arrangements of the blocks of $\mathbf{y}'_j$. By removing the last $2m$ coordinates in each $3m$-block of $\mathbf{y}'$ obtain vectors $\mathbf{y}' \sum_{j=1}^{p} \beta_j \cdot \mathbf{y}'_j \in \mathbb{Z}_q^{(2\ell+1)3m}$, and $\mathbf{y} \in \mathbb{Z}^{(2\ell+1)m}$. Now, we can declare

$$\|\mathbf{y}\|_\infty \leq \|\mathbf{y}'\|_\infty \leq \sum_{j=1}^{p} \beta_j \cdot \|\mathbf{y}_j\|_\infty = \sum_{j=1}^{p} \beta_j \cdot 1 = \beta.$$

Moreover, as $\mathbf{y}'_j \in \mathsf{SecretExt}(\bar{d})$ for all $j \in [p]$, we have that $\mathbf{y} \in \mathsf{Secret}_\beta(\bar{d})$ and, $\mathbf{A} \cdot \mathbf{y} = \mathbf{A}^* \cdot \mathbf{y}' = \mathbf{A}^* \cdot \sum_{j=1}^{p} \beta_j \cdot \mathbf{y}_j = \mathbf{A}^* (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{z,j} - \mathbf{h}_{z,j})) = \mathbf{u} \mod q$.

For each $j \in [p]$, let $\mathbf{f}'_j = (\mathbf{s}_{f,j} - \mathbf{h}_{f,j})$. Then $\phi_{f,j}(\mathbf{f}'_j) = \phi_{f,j}(\mathbf{s}_{f,j}) - \phi_{e,j}(\mathbf{h}_{f,j}) = \mathbf{a}_{f,j} \in \mathsf{B}_{3m}$, which implies that $\mathbf{f}'_j \in \mathsf{B}_{3m}$. Let $\hat{\mathbf{f}} = \sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j \in \mathbb{Z}^{3m}$ and by dropping the last $2m$ coordinates from $\hat{\mathbf{f}}$ obtain $\mathbf{f}' \in \mathbb{Z}^m$. We can declare

$$\|\mathbf{f}'\|_\infty \leq \|\hat{\mathbf{f}}\|_\infty \leq \sum_{j=1}^{p} \beta_j \cdot \|\mathbf{f}'_j\|\|_\infty = \sum_{j=1}^{p} \beta_j \cdot 1 = \beta.$$

Moreover, for each $j \in [p]$, let $\mathbf{t}'_j = (\mathbf{s}_{t,j} - \mathbf{h}_{t,j})$. Then, $\phi_{t,j}(\mathbf{t}'_j) = \phi_{t,j}(\mathbf{s}_{t,j}) - \phi_{t,j}(\mathbf{h}_{t,j}) = \mathbf{a}_{t,j} \in \mathsf{B}_{3m}$, which implies that $\mathbf{t}'_j \in \mathsf{B}_{3m}$. Let $\hat{\mathbf{t}} = \sum_{j=1}^{p} \beta_j \cdot \mathbf{t}'_j \in \mathbb{Z}^{3m}$ and by dropping the last $2m$ coordinates from $\hat{\mathbf{t}}$ obtain $\mathbf{t}' \in \mathbb{Z}^m$. We can declare

$$\|\mathbf{t}'\|_\infty \leq \|\hat{\mathbf{t}}\|_\infty \leq \sum_{j=1}^{p} \beta_j \cdot \|\mathbf{t}'_j\|_\infty = \sum_{j=1}^{p} \beta_j \cdot 1 = \beta.$$

We can obtain the relation

$$\mathbf{V}^* \cdot \hat{\mathbf{t}} + \mathbf{I}^* \cdot \hat{\mathbf{f}} = \mathbf{v} \mod q \iff \mathbf{V}^* \cdot (\mathbf{A}_{(d)} \cdot \mathbf{t}') + \mathbf{f}' = \mathbf{v} \mod q.$$

and

$$\mathbf{A}^* \cdot \sum_{j=1}^{p} \beta_j \cdot \mathbf{t}'_j = \mathbf{A}_{(d)} \cdot \mathbf{t}' = \mathbf{w} \mod q.$$

Let $d^* = \mathbf{s}_d - \mathbf{h}_d = \hat{\tau}^{-1}(\mathbf{a}_d)$. Then, it follows that $d^* \in \mathsf{B}_{2\ell}$. Now, let $d^* = (d_1, \ldots, d_\ell, d_{\ell+1}, \ldots, d_{2\ell})$ and let $d = (d_1, \ldots, d_\ell) \in 0, 1^\ell$.

For each $j \in [\bar{p}]$, let $\mathbf{e}'_j = (\mathbf{s}_{e,j} - \mathbf{h}_{e,j})$. Then $\phi_{e,j}(\mathbf{e}'_j) = \phi_{e,j}(\mathbf{s}_{e,j}) - \phi_{e,j}(\mathbf{h}_{e,j}) = \mathbf{a}_{e,j} \in \mathsf{B}_{3k}$, which implies that $\mathbf{e}'_j \in \mathsf{B}_{3k}$. Let $\hat{\mathbf{e}} = \sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{e}'_j$ and by dropping the last $2k$ coordinates from $\hat{\mathbf{e}}$ obtain $\mathbf{e}' \in \mathbb{Z}^k$. We can declare,

$$\|\mathbf{e}'\|_\infty \leq \|\hat{\mathbf{e}}\|_\infty \leq \sum_{j=1}^{\bar{p}} b_j \cdot \|\mathbf{e}'_j\|_\infty = \sum_{j=1}^{p} b_j \cdot 1 = b.$$

Now, $\|\mathbf{e}'\|_\infty \leq b$, and $\mathbf{P}^*\mathbf{e}' + \mathbf{Q}d^* = \mathbf{P}\mathbf{e}' + (0^{k-\ell} \| \lfloor q/2 \rfloor d) = \mathbf{c} \mod q$.

## References


1. Chaum, D.; Van Heyst, E. Group Signatures. In *EUROCRYPT 1991*; LNCS; Springer: Berlin/Heidelberg, Germany, 1991; Volume 547, pp. 257–265.
2. Chen, L.; Pedersen, T.P. New group signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*; LNCS; Springer: Berlin/Heidelberg, Germany, 1994; Volume 950; pp. 171–181.
3. Ateniese, G.; Tsudik, G. Group signatures á la carte. In Proceedings of the Symposium on Discrete Algorithms: Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, USA, 17–19 January 1999; Volume 17, pp. 848–849.
4. Ateniese, G.; Camenisch, J.; Joye, M.; Tsudik, G. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO 2000*; LNCS; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1880, pp. 255–270.
5. 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; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2656, pp. 614–629.
6. Bresson, E.; Stern, J. Efficient Revocation in Group Signatures. In *PKC 2001*; LNCS; Springer: Berlin/Heidelberg, Germany, 2001; Volume 1992, pp. 190–206.
7. Camenisch, J.; Lysyanskaya, A. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO 2002*; LNCS; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2442, pp. 61–76.
8. Brickell, E. *An Efficient Protocol for Anonymously Providing Assurance of the Container of the Private Key*; Submitted to Trusted Computing Group: Beaverton, OR, USA, 2003.

9. Boneh, D.; Shacham, H. Group Signatures with Verifier-Local Revocation. In Proceedings of the ACM-CCS 2004, Washington, DC, USA, 25–29 October 2004; pp. 168–177.
10. Langlois, A.; Ling, S.; Nguyen, K.; Wang, H. Lattice-Based Group Signature Scheme with Verifier-Local Revocation. In *PKC 2014*; LNCS; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8383, pp. 345–361.
11. Gordon, S.D.; Katz, J.; Vaikuntanathan, V. A Group Signature Scheme from Lattice Assumptions. In *ASIACRYPT 2010*; LNCS; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6477, pp. 395–412.
12. Camenisch, J.; Neven, G.; Rückert, M. Fully Anonymous Attribute Tokens from Lattices. In *SCN 2012*; LNCS; Springer: Berlin/Heidelberg, Germany, 2012; Volume 12, pp. 57–75.
13. Laguillaumie, F.; Langlois, A.; Libert, B.; Stehlé, D. Lattice-Based Group Signatures with Logarithmic Signature Size. In *ASIACRYPT 2013*; LNCS; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8270, pp. 41–61.
14. Ling, S.; Nguyen, K.; Wang, H. Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based. In *PKC 2015*; LNCS; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9020, pp. 427–449.
15. Boyen, X. Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In *PKC 2010*; LNCS; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6056, pp. 499–517.
16. Nguyen, P.Q.; Zhang, J.; Zhang, Z. Simpler Efficient Group Signatures from Lattices. In *PKC 2015*; LNCS; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9020, pp. 401–426.
17. 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; Springer: Berlin/Heidelberg, Germany, 2016; Volume 10032, pp. 373–403.
18. Ling, S.; Nguyen, K.; Wang, H.; Xu, Y. Lattice-Based Group Signatures: Achieving Full Dynamicity with Ease. In *ACNS 2017*; LNCS; Springer: Cham, Switzerland, 2017; Volume 10355, pp. 293–312.
19. Ishida, A.; Sakai, Y.; Emura, K.; Hanaoka, G.; Tanaka, K. Fully Anonymous Group Signature with Verifier-Local Revocation. In *SCN2018*; LNCS; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11035, pp. 23–42.
20. Perera, M.N.S.; Koshiba, T. Fully Dynamic Group Signature Scheme with Member Registration and Verifier-local Revocation. In *ICMC 2018*; PROMS; Springer: Berlin/Heidelberg, Germany, 2018; Volume 253, pp. 399–415.
21. 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; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11125, pp. 229–247.
22. Perera, M.N.S.; Koshiba, T. Achieving Strong Security and Verifier-Local Revocation for Dynamic Group Signatures from Lattice Assumptions. In *STM 2018*; LNCS; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11091, pp. 3–19.
23. Perera, M.N.S.; Koshiba, T. Achieving Strong Security and Member Registration for Lattice-based Group Signature Scheme with Verifier-local Revocation. *J. Internet Serv. Inf. Secur.* **2018**, *8*, 1–15. [CrossRef]
24. Chu, C.K.; Liu, J.K.; Huang, X.; Zhou, J. Verifier-local revocation group signatures with time-bound keys. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, Seoul, Korea, 2–4 May 2012; pp. 26–27.
25. Perera, M.N.S.; Koshiba, T. Almost-Fully Secured Fully Dynamic Group Signatures with Efficient Verifier-Local Revocation and Time-Bound Keys. In *IDCS 2018*; LNCS; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11226, pp. 134–147.
26. Perera, M.N.S.; Koshiba, T. Zero-Knowledge Proof for Lattice-Based Group Signature Schemes with Verifier-Local REVOCATION. In *NBiS 2018*; LNDECT; Springer: Berlin/Heidelberg, Germany, 2018; Volume 22, pp. 772–782.
27. Peikert, C. A Decade of Lattice Cryptography. *Found. Trends Theor. Comput. Sci.* **2016**, *10*, 283–424. [CrossRef]
28. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC 2005*; ACM Press: New York, NY, USA, 2005; pp. 84–93.
29. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *ACM STOC 08*; ACM: New York, NY, USA, 2008; pp. 197–206.
30. Ajtai, M. Generating hard instances of lattice problems. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 99–108.

31. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT 2012*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7237, pp. 700–718.
32. Agrawal, S.; Boyen, X.; Vaikuntanathan, V.; Voulgaris, P.; Wee, H. Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices. In *PKC 2012*; LNCS; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7293, pp. 280–297.
33. Cash, D.; Hofheinz, D.; Kiltz, E.; Peikert, C. Bonsai Trees, or How to Delegate a Lattice Basis. In *Eurocrypt 2010*; LNCS; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6110, pp. 523–552.
34. Naor, D.; Shenhav, A.; Wool, A. One-time signatures revisited: Have they become practical? *IACR Cryptol. Eprint Arch.* **2005**, *2005*, 442.
35. Chow, S.S.; Wong, D.S. Anonymous Identification and Designated-Verifiers Signatures from Insecure Batch Verification. In *EuroPKI 2007*; LNCS; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4582, pp. 203–219.
36. Kawachi, A.; Tanaka, K.; Xagawa, K. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *ASIACRYPT 2008*; LNCS; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5350, pp. 372–389.
37. Pointcheval, D.; Vaudenay, S. *On Provable Security for Digital Signature Algorithms*; Ecole Normale Supérieure, Laboratoire d'Informatique: Paris, France, 1996.