Article

Almost Fully Secure Lattice-based Group Signatures with Verifier-local Revocation

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

[1] Graduate School of Science and Engineering, Saitama University, Japan; perera.m.n.s.119@ms.saitama-u.ac.jp; mnisansalasperera@gmail.com

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

* Correspondence: mnisansalasperera@gmail.com; Tel.: +81-901208-6384

Abstract: Efficient member revocation and strong security against attacks are prominent requirements in group signature schemes. Among the revocation approaches Verifier-local revocation is the most flexible and efficient method since it requires to inform only the verifiers regarding the revoked members. The verifier-local revocation technique uses a token system to manage members' status. However, the existing group signature schemes with verifier-local revocability rely on weaker security. On the other hand, existing static group signature schemes rely on a stronger security notion called, full-anonymity. Achieving the full-anonymity for group signature schemes with verifier-local revocation is a quite challenging task. This paper aims to obtain stronger security for the lattice-based group signature schemes with verifier-local revocability, which is closer to the full-anonymity. Moreover, this paper delivers a new key-generation method which outputs revocation tokens without deriving from the users' signing keys. By applying the tracing algorithm given in group signature schemes for static groups, this paper also outputs an efficient tracing mechanism. Thus, we deliver a new group signature scheme with verifier-local revocation that satisfies a stronger security from lattices.

Keywords: lattice-based group signatures; verifier-local revocation; anonymity; almost-full anonymity; traceability

1. Introduction

Group Signatures, introduced by Chaum and van Heyst [1] allow the group members to issue signatures for the sake of the group while hiding their information (anonymity). On the other hand, the group manager can cancel anonymity of the signers and identify the owner of the signature (traceability). In other words, in group signature schemes, only the valid group members can sign messages, the receiver cannot identify the signer but he can authenticate any signature, and in the case of dispute an authorized person (the group manager) can identify the signer. Thus, the signer should be anonymous to the receivers and traceable to the authorities (the group manager). These two features (*anonymity* and *traceability*) make group signature schemes attractive to many real-life applications such as key-card access systems, digital right management, and anonymous printing.

Since the group signatures have been introduced, many proposals have been presented with different levels of improvements and security. Among them, the scheme presented by Chen and Pedersen [2] and the scheme suggested by Ateniese and Tsudik [3] submitted new features such as framing resistance, coalition resistance, and exculpability. Again, Ateniese et al. [4] proposed a new scheme to overcome the weaknesses of the previous schemes, and it claimed that the new scheme is the first provably secure scheme in random oracle model. Later, Bellare et al. [5] formulated a stronger security model (BMW03 model) with two security requirements called *full anonymity* and *full traceability*, which imply the existing security properties. Even though the BMW03 model is known as the strongest security model at present, it serves for static groups only. By adopting the BMW03 model, several group signatures have been proposed but constructing a scheme which supports revocation and with a high-security level is a challenge.

1.1. Member revocation approaches

In real world, almost all the group settings are stateless. Member revocation is one of the principal features of a group. Misbehaved and retired members should be restricted generating signatures on behalf of the group in future. One naive approach for member revocation is replacing all the keys newly except for the revoking member when a member is revoked. Thus any revoked member cannot produce a valid signature because he does not know the new keys. But since this approach requires to distribute the newly generated keys to all the members, verifiers, and authorized persons, this is not suitable for large groups. Bresson and Stern [6] proposed a method which requires signers to prove at the time of signing that his member certificate is not in the public revocation list. Camenisch et al. [7] suggested a revocation method that used dynamic accumulators (*accumulator* is an algorithm, that allows hashing a large set of inputs to one shorter value and *dynamic accumulator* allows to add or delete inputs dynamically). Since the approach proposed by Camenisch et al. [7] requires members to keep track of revoked user information, and needs to update their membership for each time that a member is revoked, workloads of the existing group members increase.

A different approach called *Verifier-local Revocation (VLR)* was suggested by Brickell [8] and formalized by Boneh et al. [9] in their group signature scheme. In VLR mechanism, every member has a revocation token other than their secret signing key to identify his status, i.e., whether he is revoked or not. When a member is revoked his token is placed on a list called *revocation list (RL)* and the latest revocation list is passed to the verifiers. The verifiers can use this revocation list at the time of verifying a signature to authenticate whether the signer is an existing member or not. Since the number of verifiers in a group is less than the number of members, VLR mechanism is convenient for large groups than any other revocation approaches. Because of these reasons, VLR is considered as the most flexible revocation method at present.

In general, group signature schemes consist of four algorithms, namely, KeyGen, Sign, Verify, and Open. On the other hand, any VLR group signature scheme consists of only the former three algorithms because it has an *implicit tracing algorithm* instead of Open for tracing misbehaved users.

The implicit tracing algorithm executes Verifiy repeatedly for each user until it returns invalid. Then the implicit tracing algorithm returns the index of the first user for which Verifiy returns invalid. The returned index is the index of the misbehaved user. For a given signature, the implicit tracing algorithm can trace at least one user who generated the signature.

Most of the VLR group signature schemes follow the bilinear map setting, which will be insecure when quantum computers become a reality. Lattice-based cryptography is one of the answers for post-quantum and the first lattice-based group signature scheme with VLR was proposed in 2014 by Langlois et al. [10].

1.2. Lattice-based Group Signature Schemes

At present lattice-based cryptography is the most important candidate for post-quantum cryptography because it holds a great promise against quantum computers. Lattice-based cryptography has strong security proofs based on the worst-case hardness of the lattice problems and efficient implementation.

The first lattice-based group signature scheme was proposed by Gordon et al. [11] in 2010. A noticeable disadvantage of this scheme is the linear barrier. i.e., the size of the group signature increases with the number of members N in a group. Thus the size of the signatures given in the scheme in [11] is $\mathcal{O}(N)$ Then Camenisch et al. [12] proposed a more secure and efficient scheme with an anonymous attribute token system. However, Camenisch's scheme [12] was also unable to overcome the linear-size problem as its signature size is still linear in N. Finally, the linear-size problem was overcome by the scheme proposed by Languillaumie et al. [13]. In their scheme, the group public key size and the signature size are both proportional to $\log N$.

However, above three lattice-based group signature schemes support only static groups, not dynamic groups. In 2014, Langlois et al. [10] suggested the first lattice-based group signature scheme that supports member revocation. In their scheme they have used *Verifier-local Revocation (VLR)* as the revocation mechanism and thus the first revocation scheme that is believed to be quantum-resistant at that time. Moreover, their scheme has several advantages over previously proposed schemes. For instance, the scheme given in [10] is simple as the signature of it is basically an all-in-one proof of knowledge. Further, it has shorter signatures and group public keys comparing to other schemes. Even though this scheme has several remarkable advantages over the previous works, the security of the scheme is weaker since the scheme satisfies a relaxed security notion called *selfless-anonymity*. Moreover, like any other VLR group signature scheme it has the implicit tracing algorithm that uses revocation token as the tracing key. Since the implicit tracing algorithm requires to run Verify until the algorithm returns invalid, this is not suitable for large groups.

Most of the schemes proposed after 2003 use the BMW03 model, which is known as the strongest security model at present. Among those schemes, the lattice-based group signature scheme suggested by Ling et al. [14] has achieved significant features than the other schemes. The scheme [14] relies on relatively weak security assumptions and the size of the public key and the signature is shorter. Moreover, the scheme itself simpler since the construction is based on Boyen's signature scheme [15] which is known as a simple construction. Addition to these, the scheme in [14] has a ring variant which is considered as a noticeable approach. However, this scheme satisfies only static groups, not dynamic groups because the scheme does not support member registration or member revocation.

Later, Nguyen et al. [16] also proposed a simpler group signature scheme. In their scheme the security is reduced to the hardness of Short Integer Solution (SIS) and Learning With Errors (LWE) in the random oracle model. However, this scheme is also available for static groups not for dynamic groups.

Libert et al. [17] constructed a group signature scheme based on lattice assumptions for dynamic groups. In their work, they have facilitated the user registration but have not considered the user revocation problem. The first fully dynamic group signature scheme from lattices was proposed by Ling et al. [**?**] using accumulators, which seems to be less efficient than using VLR in large groups.

1.3. Our Contribution

In our work, we focus on presenting a lattice-based group signature scheme which supports efficient member revocation and which is with a high security level. When applying a member revocation mechanism, we consider an approach which will not require changing the existing keys when a member is revoked. Thus, *Verifier-local Revocation* (VLR) seems to be the most flexible revocation approaches for our scheme, because it only requires to update the verifiers with revocation information, but not the existing members at the member revocation. We adopt features in the BMW03 model to make our scheme secure. However, since the previous VLR schemes have not use the BMW03 model and the BMW03 model was not proposed for dynamic groups, we have to cope with revocation queries (in the anonymity game) in the BMW03 model. The revocation query allows the adversary to request for the revocation tokens of the members. However, if the adversary obtains the challenged members' tokens, he can identify the challenged signature's index by executing Verify with the challenged members' tokens. Thus, coping with revocation query in the BMW03 model is quite difficult. The scheme given in [18] which is based on general assumptions, provides member revocation with VLR and satisfies a stronger security than the security given in the original VLR group signature schemes. The scheme given in [18] suggested a security notion called *almost-full anonymity*, which is a restricted version of full anonymity given in the BMW03 model. The almost-full anonymity give all the secret signing keys to the adversary as the full-anonymity game and allow revocation query as an additional feature. However, it will not allow the adversary to access revocation token related to the challenging incides, and it will not generate the challenging signature for the indices which are used in the revocation queries. The scheme given in [18] is based on general assumption. Thus, we discuss how to employ the almost-full anonymity for lattice-based VLR group signature schemes. Thus, we use the almost-full anonymity to secure our lattice-based group signature scheme with VLR.

In the VLR group signature scheme given in [10] revocation tokens are part of the particular secret signing keys. Since we are providing all the secret keys to the adversary at the anonymity game, the adversary can create revocation tokens using the information he 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, we deliver a different method to generate revocation tokens in our scheme.

The implicit tracing algorithm given in VLR group signature schemes are not suitable for a large groups because the time consumption is high in the implicit tracing algorithm. Thus we use the explicit tracing algorithm for the lattice-based group signature scheme with VLR to identify any user. Accordingly, we use the group manager's secret key to find the signers instead of running Verify a linear time in the number of users as in previous VLR schemes with the implicit tracing algorithm. Since the explicit tracing algorithm, helps to identify any signer by running it only once, this can be used for any large groups. As a result, we propose a lattice-based group signature scheme, which is almost fully secured, supports member revocation and tracing signers efficiently, which provides a new method to generate revocation tokens, and suitable even for a large group.

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 our lattice-based group signature scheme 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 5. In Section 6 we conclude the paper and discuss open problems.

2. Preliminaries

2.1. Notations

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

A negligible function, denoted by $negl(n)$, is an $f(n)$ such that $f(n) = o(n^{-c})$ for every fixed constant c. Moreover, the *statistical distance* between two distributions X and Y over a countable domain D is $\frac{1}{2}\sum_{d \in D} |X(d) - Y(d)|$ and we say two distributions (formally, two ensembles of distributions indexed by n) are *statistically close* if their statistical distance is negligible in n.

We denote by n the security parameter. The maximum number of expected users in a group is $N = 2^{\ell}$ and each member's identity is denoted by a string $d \in \{0,1\}^{\ell}$, which is the binary representation of his index. Depending on the given n we fix the other parameters as in Table 1.

Table 1. Parameters of the scheme

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

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

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

2.2. Lattices

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

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

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

We consider a discrete Gaussian distribution for a lattice. The Gaussian function centered in a vector $\mathbf{c}$ with parameter $s > 0$ is defined as $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2}$ and the corresponding probability density function proportional to $\rho_{s,\mathbf{c}}$ is defined as $D_{s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/s^n$ for all $\mathbf{x} \in \mathbb{R}^n$. The discrete Gaussian distribution with respect to a lattice Λ is defined as $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = D_{s,\mathbf{c}}(\mathbf{x})/D_{s,\mathbf{c}}(\Lambda) = \rho_{s,\mathbf{c}}(\mathbf{x})/\rho_{s,\mathbf{c}}(\Lambda)$ for all $\mathbf{x} \in \Lambda$. Since $\mathbb{Z}^m$ is also a lattice, we can define a discrete Gaussian distribution for $\mathbb{Z}^m$. By $D_{\mathbb{Z}^m,\sigma}$, we denote the discrete Gaussian distribution for $\mathbb{Z}^m$ around the origin with the standard deviation σ.

2.3. Lattice-Related Properties

Here we describe the hardness of computational problems of lattices that we use in our scheme. First we define SIVP problem. Then we define 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 ($SIVP_\gamma$ [19]). *Given a basis $\mathbf{B}$ of an n-dimentional lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, finding linearly independent vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n$ is $SIVP_\gamma$ problem, where $||\mathbf{s}_i|| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$ for all i ($\lambda_n(\mathcal{L})$ is n-th successive minimum).*

2.3.2. Learning With Errors (LWE)

Regev [20] introduced LWE problem, which is a lattice problem and hard to solve. His work is to result a reduction from worst-case lattice problems to a certain learning problem.

Definition 2 (Learning With Errors Problem ($\mathbf{LWE}_{n,q,\chi}$) [19])**.** *LWE is parametrized by $n, m \geq 1, q \geq 2$, and χ. For $\boldsymbol{s} \in \mathbb{Z}_q^n$, the distribution $\mathrm{A}_{\boldsymbol{s},\chi}$ is obtained by sampling $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random and $\mathrm{e} \leftarrow \chi$, and outputting the pair $(\boldsymbol{a}, \boldsymbol{a}^T \cdot \boldsymbol{s} + \mathrm{e})$.*

There are two version of LWE problem: *Search-LWE* and *Decision-LWE*. Search-LWE is to find the secret **s** and Decision-LWE is to distinguish LWE samples and samples chosen according to the uniformly distribution. We use the hardness of Decision-LWE problem for our scheme.

For a prime power q, $\beta \geq \sqrt{n}\omega(\log n)$, and distribution χ, solving $LWE_{n,q,\chi}$ problem is at least as hard as solving $SIVP_\gamma$, where $\gamma = \tilde{O}(nq/\beta)$ [20,21].

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

SIS was first introduced in seminal work of Ajtai [22]. SIS has served in many applications as identification schemes, one-way hash functions and digital signatures. SIS problem asks to find a sufficiently short nontrivial integer combination of given uniformly random elements of a certain large finite additive group, which sums to zero [19].

Definition 3 (Short Integer Solution Problem ($\mathbf{SIS}_{n,m,q,\beta}$) [19,20])**.** *Given m uniformly random vectors $\boldsymbol{a}_i \in \mathbb{Z}_q^n$, forming the columns of a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, find a nonzero vector $\boldsymbol{x} \in \Lambda^\perp(\boldsymbol{A})$ such that $||\boldsymbol{x}|| \leq \beta$ and $\boldsymbol{Ax} = 0 \mod q$.*

SIS problem is for *homogeneous* systems. Later, Gentry et al. [21] formalized its *inhomogeneous* version ISIS problem.

Definition 4 (Inhomogeneous Short Integer Solution Problem ($\mathbf{ISIS}_{n,m,q,\beta}$) [21])**.** *Given matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with m uniformly random vectors $\boldsymbol{a}_i \in \mathbb{Z}_q^n$ and a uniformly random vector $\mathbf{y} \in \mathbb{Z}_q^n$, $\mathbf{ISIS}_{n,m,q,\beta}$ asks to find a nonzero vector $\boldsymbol{x} \in \Lambda_{\boldsymbol{u}}^\perp(\boldsymbol{A})$ such that $||\mathbf{x}|| \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \mod q$.*

For any m, β, and for any $q \geq \beta \cdot \omega(\sqrt{n \log n})$, solving $SIS_{n,m,q,\beta}$ problem and $ISIS_{n,m,q,\beta}$ problem with non-negligible probability is at least as hard as solving $SIVP_\gamma$ problem, for some $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ [21].

2.4. Lattice-Related Algorithms

For our construction, we require a family of functions such that each function capable of compute with any input but not feasible to invert the given input. Such family of functions are called *one-way functions*. *Trapdoor functions* are one-way functions that keep a secret information (trapdoor) and without this information finding the inverse of the function is hard. We use trapdoor functions in our constructions since anyone cannot identify the inverse of the function without the trapdoor.

We use a randomized nearest-plane algorithm called SampleD, which was discussed in [21] and [23]. The algorithm SampleD samples from a discrete Gaussian $D_{\Lambda,s,\mathbf{c}}$ over any lattice Λ. We use the version given in [23].

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

According to [21] the inputs to SampleD is an (ordered) basis B of an n-dimensional lattice Λ, a parameter $s > 0$, and a center c and SampleD always outputs a lattice vector.

The notion of preimage sampleable trapdoor functions (PSTFs) was discussed in [21]. PSTFs are defined by probabilistic polynomial-time algorithms GenTrap, SamplePre. There are several constructions of PSTFs. We use GenTrap and SamplePre given in [23,24].

- GenTrap(n, m, q) is an efficient randomized algorithm that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor matrix $\mathbf{R}$ for a given any integers $n \geq 1, q \geq 2$, and sufficiently large $m \geq 2n \log q$. The distribution of the output $\mathbf{A}$ is negl(n)-far from the uniform distribution.
- SamplePre($\mathbf{A}$, $\mathbf{R}$, $\mathbf{u}$, σ) outputs a sample $\mathbf{e} \in \mathbb{Z}^m$ from a distribution that is within negligible statistical distance of $D_{\Lambda_q^{\mathbf{u}}(A),\sigma}$, on input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a trapdoor basis $\mathbf{R}$, a target image $\mathbf{u} \in \mathbb{Z}_q^n$, and the standard deviation $\sigma = \omega(\sqrt{n \log q \log n})$.

Moreover, we use *witness decomposition and extensions* technique (WitnessDE) and *matrix extension* technique (MatrixExt) described in [10]. These techniques are needed to convince the verifier the prover's witness in the interactive protocol discussed in Section 4.

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

  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 outputs 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 generated by appending *2m zero-columns* to each of the component-matrices of $\mathbf{A}$.

2.5. VLR Group Signature

VLR group signatures require distributing the revocation list only to the verifiers when a member is revoked. While VLR group signatures are for dynamic groups, the general group signature scheme is for static groups and consists of four probabilistic polynomial time (PPT) algorithms: KeyGen, Sign, Verify and Open [5].

- KeyGen(n,N): This randomized PPT algorithm outputs a group public key **gpk**, a group manager secret key **gmsk**, and user secret keys **gsk**[d] ($d \in \{0,\ldots,N-1\}$) for the security parameter n and the number of group members N.
- Sign(**gpk**, **gsk**[d], M): This randomized algorithm is used by group members to generate a signature Σ on a message M using the group public key **gpk** and user secret key **gsk**[d].
- Verify(**gpk**, Σ, M): This deterministic algorithm verifies whether the generated signature Σ is valid on the given message M using **gpk**.
- Open(**gmsk**, M, Σ): This algorithm takes as inputs the group manager's secret key **gmsk**, a message M, and a signature Σ on M and returns the index of the user, who has generated the signature. If Open cannot find the signer, then it returns the failure symbol.

VLR group signature schemes consist of three PPT algorithms [9] since the *implicit tracing algorithm* is used to trace the misbehaved users.

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

Implicit Tracing Algorithm: Any VLR group signature scheme has an *implicit tracing algorithm*, that uses **grt** as the tracing key. For a given valid message-signature pair (M, Σ), authorized person who knows all the tracing keys **grt** can execute Verify(**gpk**, RL=**grt**[i], Σ, M) for $i = 0, 1, \ldots, N-1$ until Verify returns $Invalid$. The first index $i^* \in \{0, 1, \ldots, N-1\}$ for which Verify returns $Invalid$ is the index of the signer. The tracing algorithm fails if this algorithm verifies properly for all users on the given signature. Since the implicit tracing algorithm requires to run Verify linear times in N, it is inappropriate for a large group because to detect a single user the group manager has to check almost all users until he finds the user who generated the particular signature. In comparison to the algorithm Open, time consumption of the implicit tracing algorithm is high.

2.6. Other Tools

The interactive protocol, which is described in Section 4 is the main building block of our scheme. It allows the user to convince the verifier that he is a certified group member and not being revoked.

We construct our scheme based on the construction of the scheme given in [10]. Hence, 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 other than their secret signing keys to confirm their validity to sign messages. At the verification stage, *Revocation List*, which is denoted by RL is given as an additional parameter to the verification algorithm. RL contains all the revocation tokens of the revoked users. Thus, the verifier can use RL to verify the validity of the user, who signed the message.

In the construction, we use an one-time signature scheme $\mathcal{OTS}$ = (OGen, OSign, OVer). OGen is the key generation algorithm, OSign is the signing algorithm, and OVer is the verification algorithm [25]. $\mathcal{OTS}$ schemes are based on one-way functions, which are simpler to implement and are computationally efficient than trapdoor functions. $\mathcal{OTS}$ schemes are digital signature schemes, which require the signer to generate keys for each message to be signed. As a result, keys generated for each message are unique for the particular messages. OGen outputs a signing / verification key pair (**osk**, **ovk**) for input (1^n). OSign takes **osk** and a message M as inputs and outputs a signature Σ. OVer is a deterministic algorithm that takes **ovk**, the message M, and the signature Σ as inputs to validate the signature Σ. Depending on the signature validation it outputs valid $\top$ or invalid $\perp$ [26].

3. Definitions of the Security Notations

This section first discusses the existing security notions for anonymity. Then it explains difficulties of achieving full-anonymity for VLR schemes with revocation query and defines the almost-full anonymity. Finally, traceability is declared.

Since Chaum and van Heyst [1] introduced group signatures, more security properties have been considered according to the requirements of different 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 speaking, anonymity and traceability mean are specified as the following.

- *Anonymity* requires that no adversary recovers the identity of the user from its signature, which is generated by one of the indices from two indistinguishable indices.
- *Traceability* requires that no adversary forges a signature that cannot be traced.

In 2003 Bellare et al.[5] developed a standard security model (BMW03 model) for group signatures with two security properties, *full anonymity* and *full traceability*, which implies the existing unformalized requirements. In 2004 Boneh et al.[9] proposed a relaxed anonymity called *selfless-anonymity*, which is weaker in security in their scheme with VLR.

3.1. Anonymity

- *Full anonymity* allows the adversary to corrupt all the users of the group, and the adversary can access the opening oracle to make queries. In the beginning, all the user secret keys and the public keys are given to the adversary, and he can obtain the outcome of the algorithm Open for any signature as he wishes.
- *Selfless-anonymity* is a relaxed anonymity and it differs from the full-anonymity by the limitations it has. The selfless-anonymity provides none of the user secret keys to the adversary, but only the public keys at the beginning. However, even with these weaknesses, *selfless-anonymity* facilitates any user to determine whether his secret key is used to generate a particular signature if he forgets whether he signed the message. This CCA-anonymity notation, the selfless-anonymity allows three types of queries: *Signing*, *Corruption*, and *Revocation*.

The anonymity game between a challenger and an adversary in selfless-anonymity is as follows. The adversary in selfless-anonymity game is weaker than the adversary in full anonymity game because the adversary in the selfless-anonymity game has not given access to any secret key. The adversary has to determine the key which is used to generate the signature in this game.

- **Initial Phase:** The challenger C runs KeyGen algorithm to obtain a group public key **gpk**, group members' secret keys **gsk** and tokens **grt**. Then gives **gpk** to the adversary A.
- **Query Phase:** The adversary A can do the following queries.
  1. Signing: The adversary A requests a signature for any message M with any user index i, and the challenger C returns $\Sigma = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i], M)$.
  2. Corruption: The adversary A queries for the secret key of any user i, and the challenger C returns $\mathbf{gsk}[i]$.
  3. Revocation: The adversary A queries for the revocation token of any user i, and the challenger C returns $\mathbf{grt}[i]$.
- **Challenge Phase:** The adversary A outputs 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. The Challenger C selects a bit $b \xleftarrow{\$} \{0,1\}$, computes signature Σ^* of user i_b using $\mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i_b], M)$, and sends the challenging signature Σ^* to the adversary A.
- **Restricted Queries:** Even after the challenge phase the adversary A can do the queries but with following restrictions.
  1. Signing: The adversary A can do this query as before.
  2. Corruption: The adversary A cannot query for i_0 or i_1.
  3. Revocation: The adversary A cannot query for i_0 or i_1.
- **Guessing Phase:** Finally, the adversary A outputs a bit b', the guess of b. If $b' = b$, then A wins.

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

The first lattice-based VLR group signature scheme relies 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 in the 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 included in the previous full anonymity games given in the BMW03 model or in other schemes used the BMW03 model such as [14]. Our scheme is for dynamic-groups, which supports member revocation. Since we wish to make our VLR group signature scheme full-anonymous, we will concern a new security notion for "full anonymity with revocation query". But here we have to concern about the risk of giving revocation tokens to the adversary. Simply adding the revocation query given in the selfless-anonymity to the notion full anonymity will make our scheme insecure. The definition of full anonymity with revocation query is as below.

- **Initial Phase:** The challenger C runs the algorithm KeyGen to obtain a group public key **gpk**, a group manager's secret key **gmsk**, and group members' secret keys **gsk** and revocation tokens **grt**. Then gives (**gpk**, **gsk**) to the adversary.
- **Query Phase:** The adversary A can query any token (**grt**) of any user and A can access the opening oracle, which results with $\mathsf{Open}(\mathbf{gmsk}, M, \Sigma)$ when A queried with any message M and a valid signature Σ.
- **Challenge Phase:** The adversary A outputs a message M and two distinct identities i_0, i_1. The challenger C selects a bit $b \overset{\$}{\leftarrow} \{0,1\}$, generates $\Sigma^* = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[i_b], \mathbf{grt}[i_b], M)$ and sends Σ^* to the adversary A. The adversary still can query the opening oracle except the signature challenged but he is not allowed for revocation queries.
- **Guessing Phase:** Finally, adversary A outputs a bit b', the guess of b. If $b' = b$, then the adversary A 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 has and guess the index of the generated signature. The advantage of A in winning the game is $Adv_A = |\Pr[b' = b] - 1/2|$. Since the adversary can obtain the tokens of the challenged indices he can win the game easily. Thus, Adv_A is not negligible.

In such away, 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 our scheme none secured.

Because of this problem, we have to concern a security notion which will not provide all the revocation tokens to the adversary. Thus, we use the almost-full anonymity given in [18], 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 the challenger C generates the keys, and both **gpk** and **gsk** are given to the adversary A as the existing full anonymity game. The adversary A can query the opening oracle with any group signature of his choice on a message M, and the oracle returns $\mathsf{Open}(\mathbf{gmsk}, M, \Sigma)$ as usual. In addition to that, the adversary A can query for the token of any user d, and the challenger returns $\mathbf{grt}[d]$. This is the revocation query suggested to the full anonymity as the additional query, which was not in the original notion of full anonymity. Then adversary A outputs two valid identities i_0, i_1 with a message M. The challenger C selects one of the two identities, which are not being queried before in revocation query phase and outputs Σ^*. Here signatures are not generated for the indices that have been queried for revocation tokens since the adversary A can use the tokens and execute Verify to check the generated signature. The adversary's goal is to determine the identity that is used to generate Σ^*. He still can query the opening oracle except for the challenged signature, and he can request revocation token of any user except the challenging indices. The almost-full anonymity game between a challenger and an adversary is as below.

- **Initial Phase:** The challenger C runs the algorithm KeyGen to obtain a group public key **gpk**, a group manager's secret key **gmsk**, and group members' secret keys **gsk** and revocation tokens **grt**. Then gives (**gpk**, **gsk**) to the adversary.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 August 2018 doi:10.20944/preprints201808.0014.v1

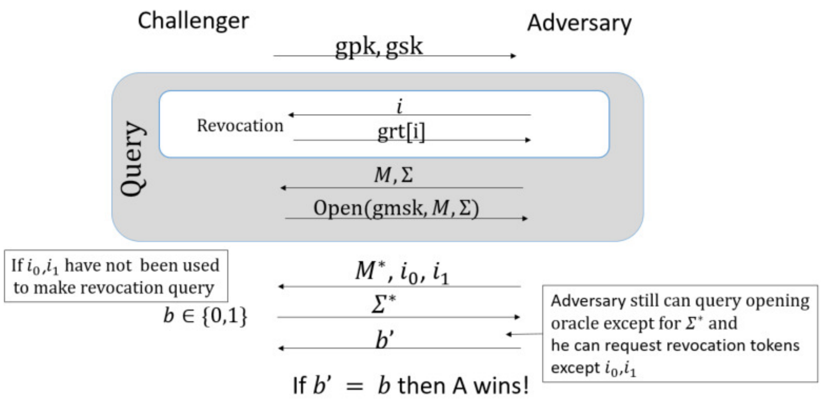
Challenger
Adversary
gpk, gsk
Query
Revocation
i
grt[i]
M, Σ
Open(gmsk, M, Σ)
If i_0,i_1 have not been used to make revocation query
$b \in \{0,1\}$
M^*, i_0, i_1
Σ^*
b'
Adversary still can query opening oracle except for Σ^* and he can request revocation tokens except i_0,i_1
If $b' = b$ then A wins!

Figure 1. Almost-full anonymity

- **Query Phase:** The adversary A can query any token (**grt**) of any user and A can access the opening oracle, which results with Open(**gmsk**, M, Σ) when A queried with any message M and a valid signature Σ.
- **Challenge Phase:** The adversary A outputs a message M and two distinct identities i_0, i_1, such that A never queried the tokens of them. The challenger C selects a bit $b \overset{\$}{\leftarrow} \{0,1\}$, generates Σ^*=Sign(**gpk**, **gsk**$[i_b]$, **grt**$[i_b]$, M), and sends Σ^* to the adversary A. The adversary still can query the opening oracle except the signature challenged, and he can request revocation tokens of any user except the indices used for challenge.
- **Guessing Phase:** Finally, adversary A outputs a bit b', the guess of b. If $b' = b$, then the adversary A wins.

We define 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.

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 generated, they are not provided to A in initial phase. In 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 outputs i_0, i_1, which are not used in the revocation query as in the selfless-anonymity. Still A can access Open but not the signature challenged and he is allowed for further revocation queries except for indices challenged. Thus, the almost-full anonymity is stronger than the selfless-anonymity since all the secret signing keys are provided to the adversary. But the almost-full anonymity is not strong as the full anonymity because all the revocation tokens are not given to the adversary. However, all the secret signing keys are given 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 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 restricted version of the full anonymity and it is somewhat weaker than the full anonymity. To be full anonymous all the secret keys (both secret signing keys and revocation tokens) should be given to the adversary at the beginning of the game. Anyhow, the almost-full anonymity is a reasonable solution for our scheme rather than the selfless-anonymity.

The VLR group signature scheme in [10] generates revocation tokens **grt** by taking a part of the secret keys **gsk**. Since we are providing all the secret keys to the adversary, and he can query revocation tokens, he can create challenged indices' tokens using the secret keys he has. Thus, we generate the revocation tokens in a different way as discussed in Section 4.

3.4. Traceability

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

In the traceability game, the adversary's challenge is to forge a signature that cannot be traced. Any group signature scheme is traceable if no adversary can win this challenge. Hence 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:** The challenger C runs KeyGen to obtain a group public key **gpk**, group members' secret keys **gsk**, and tokens **grt**. Then gives (**gpk**, **grt**) to the adversary A and sets corruption list $U \leftarrow \emptyset$.
- **Query Phase:** The adversary A can do the following queries.
  1. Signing: The adversary A requests a signature for any message M for any user index i and the challenger C returns Σ = Sign(**gpk**,**gsk**[i],M).
  2. Corruption: The adversary A queries for the secret key of any user i. The challenger C adds i to U, and returns **gsk**[i].
- **Challenge Phase:** The adversary A outputs a message M^*, a set of revocation tokens RL^*, and a signature Σ^*.
- The forgery adversary A wins if the followings are true.
  1. Σ^* is accepted as a valid signature on the message M^* with RL^*.
  2. Σ^* traces to some user outside the coalition $U \backslash RL^*$ or tracing algorithm fails.
  3. Σ^* is not obtained 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.

The full traceability game between a challenger and an adversary is as below.

As explained in [5] and in [14] the group public key **gpk** and the group manager's secret key **gmsk** is given to the adversary A at the beginning of the game, and the adversary A makes queries as the following game.

- **Initial Phase:** The challenger C runs KeyGen to obtain a group public key **gpk**, a group manager's secret key **gmsk**, and group members' secret keys **gsk**. Then gives **gpk** and **gmsk** to the adversary A and sets $U \leftarrow \emptyset$.
- **Query Phase:** The adversary A can do the following queries.
  1. Signing: The adversary A requests a signature for any message M for any user index i and the challenger C returns Σ = Sign(**gpk**,**gsk**[i],M).
  2. Corruption: The adversary A queries for the secret key of any user i. The challenger C adds i to U and returns **gsk**[i].
- **Challenge Phase:** The adversary A outputs a message M^* and a signature Σ^*.

- The forgery adversary A wins if the followings are true:
  1. Σ^* is accepted as a valid signature on the message M^*.
  2. Σ^* traces to some user outside the coalition U or tracing algorithm fails.
  3. Σ^* is not obtained 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 full-traceable if Adv_A^{trace} is negligible for all polynomial N and for all polynomial time adversaries A.

4. Our VLR Scheme

In this section, first, we describe the underlying zero-knowledge interactive protocol system, which can be used by any signer to prove his validity. Then we discuss the adoption of the *explicit tracing algorithm* to our VLR scheme to trace the signer of any signature efficiently. Later in this section, we present our lattice-based group signature scheme which supports member revocation.

In our scheme, we generate member revocation tokens as a combination of part of the public key and some randomness. Even 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 since we are giving all the secret keys to the adversary at 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 a random vector.

Our scheme consists of four algorithms as below.

- KeyGen(n,N): This randomized PPT algorithm on inputs the security parameter $n \in \mathbb{N}$ and the number of group users N outputs a group public key **gpk**, a group manager secret key **gmsk**, a vector of user secret keys **gsk** = $(\mathbf{gsk}[0], \mathbf{gsk}[1], \ldots, \mathbf{gsk}[N-1])$, and a vector of user revocation tokens **grt** = $(\mathbf{grt}[0], \mathbf{grt}[1], \ldots, \mathbf{grt}[N-1])$, where $\mathbf{gsk}[i]$ is the i-th user's secret key and $\mathbf{grt}[i]$ is his revocation token.
- Sign(**gpk**, **gsk**[d], **grt**[d], M): This randomized algorithm takes as inputs a secret signing key **gsk**[d], revocation token **grt**[d], the group public key **gpk** and a message $M \in \{0,1\}^*$, and computes a group signature Σ on M.
- Verify(**gpk**, RL, Σ, M): This algorithm determines whether the given Σ is a valid signature using the given group public key **gpk** and the message M. Then it validates the signer not being revoked using RL.
- Open(**gmsk**, M, Σ): This algorithm takes as inputs the group manager's secret key **gmsk**, a message M, and a signature Σ on M, and returns the index of the user who has generated the signature. It returns failure symbol when the user cannot be identified.

4.1. The Underlying Zero Knowledge Interactive Protocol

Zero Knowledge Interactive Protocol is the building block of this scheme as it allows a prover (signer) to argue that he is a certified group user who has a valid secret key.

Let COM be the statistically hiding and computationally binding commitment scheme described in [27].

Our scheme can be seen as an adaptation of [10] and [14]. In [28] combined interactive protocol is given that signer can prove his validity of signing, his revocation token is not in the revocation list, and his index is correctly encrypted. Thus, the interactive protocol given in [28] suits for our scheme well.

We use matrix $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$, vector $\mathbf{u} \in \mathbb{Z}_q^n$, matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, 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 $\mathbf{c} \in \mathbb{Z}_q^{k_1}$ as the public parameters. The witness of the prover consists of 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 $\mathbf{e} \in \mathbb{Z}^{k_2}$. While keeping prover's identity d in secret, he has to convince the verifier that $\mathbf{A}\cdot\mathbf{x} = \mathbf{u} \bmod q$, $||\mathbf{e}_1||_\infty \leq \beta$ and $\mathbf{V}\cdot(\mathbf{B}\cdot\mathbf{r}) + \mathbf{e}_1 = \mathbf{v} \bmod q$. Moreover, the prover has to show his identity is correctly encrypted, such that $||\mathbf{e}||_\infty \leq b$ and $\mathbf{Pe} + (0^{k_1-\ell}||\lfloor q/2\rfloor d) = \mathbf{c} \mod q$.

We recall the underlying zero-knowledge protocol given in [28] in Appendix A.

As discussed in [28] we construct an efficient simulator S interacting with a (probably dishonest) verifier $\hat{\mathcal{V}}$, such that, given only the public input S outputs with probability negligibly apart from $2/3$ a simulated transcript that is statistically close to the one produced in the real interaction by the honest prover. Thus, the simulator can successfully imitate the honest prover with probability negligibly far from $2/3$.

4.2. Explicit Tracing Algorithm

In general, VLR group signature scheme has the implicit tracing algorithm, which requires to run Verify linear in the number of users until the signer is traced. Because of this reason, the implicit tracing algorithm is inconvenient for large groups, and we use the explicit tracing algorithm while using some methods described in [14] to make our scheme flexible for any group size. In general, when the explicit tracing algorithm is used, the group manager's or some other third party authorities' involvement is required, and his secret key is used to decrypt the signature on a message to obtain the index of the signer. Since the scheme in [10] consists of the implicit tracing algorithm, it does not concern the group manager's key or encryption and decryption of the identity of the signer as in [14], where the explicit tracing algorithm is used. Thus, we focus on generating the group manager's secret key and encryption and decryption of the identity of the signer by using the techniques given in [14].

4.3. Description of Our Scheme

In this section, we describe the algorithms of our scheme with *explicit tracing algorithm* Open.

Let n be the security parameter, and N be the maximum expected number of group members (users) of the group.

Key Generation: This randomized algorithm KeyGen(n,N) proceeds through the following steps to generate a group public key **gpk**, group user secret keys **gsk**, group user tokens **grt**, and a group manager secret key **gmsk**.

1. Run PPT algorithm GenTrap(n, m, q) to obtain a matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{R}$.
2. Sample a vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.
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 public parameter, the matrix $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$.
5. Run the algorithm GenTrap(n,m,q) to obtain $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and trapdoor $\mathbf{T_B}$.
6. For each user $d \in \{0,1,\cdots,N-1\}$, secret keys **gsk[**d**]** and revocation tokens **grt[**d**]** are generated as follows.
   (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 $\mathbf{x}_0 \in \mathbb{Z}^m \leftarrow$ SampleD($\mathbf{R}, \mathbf{A}_0, \mathbf{u}-\mathbf{z}, \sigma$).
   (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 $||\mathbf{x}^{(d)}||_\infty \leq \beta$, then proceed else repeat from (a).
   (g) Let the user secret key be **gsk[**d**]** = $\mathbf{x}^{(d)}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 August 2018 doi:10.20944/preprints201808.0014.v1

(h) Sample a vector $\mathbf{r}_1 \leftarrow D_{\mathbb{Z}^m,\sigma}$.
(i) Get $\mathbf{r}_2 \in \mathbb{Z}^m \leftarrow \mathsf{SampleD}(\mathbf{T_B}, \mathbf{B}, \mathbf{u} - \mathbf{z}, \sigma)$.
(j) Compute $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$.
(k) Let the user revocation token be $\mathbf{grt}[d] = \mathbf{B} \cdot \mathbf{r}$.

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

Signing: We use the one-time signature scheme $\mathcal{OTS}$ to generate keys and signatures to make our signature secure. Then we use the *zero knowledge interactive protocol* to show that the user is valid. The randomized algorithm Sign(**gpk**, **gsk**[d], **grt**[d], M) takes as inputs the group public key **gpk**, the user secret signing key **gsk**[d], revocation token **grt**[d], and generates Σ on a message M as follows.

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

1. Run $\mathsf{OGen}(1^n)$ to obtain a key pair (**ovk, osk**).
2. Encrypt the index d as follows.

   (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 $\rho \xleftarrow{\$} \{0,1\}^n$, let $\mathbf{V} = \mathcal{G}(\mathbf{A}, \mathbf{u}, \mathbf{B}, M, \rho) \in \mathbb{Z}_q^{m\times n}$.
4. Compute $\mathbf{v} = \mathbf{V} \cdot (\mathbf{B} \cdot \mathbf{r}) + \mathbf{e}_1 \mod q$ $(||\mathbf{e}_1||_\infty \leq \beta$ with overwhelming probability and $\mathbf{B} \cdot \mathbf{r} = \mathbf{grt}[d])$.
5. Generate the parameters for the interactive protocol to show the index d is encrypted correctly as follows.

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

6. Repeat the zero knowledge interactive protocol 4.1 of the commitment described above $t = \omega(\log n)$ times with the public parameter (**A**, **u**, **B**, **V**, **v**, **P**, **c**) and prover's witness (**x**, **r**, $\mathbf{e}_1$, **e**) to make the soundness error negligible and prove that user is certified. Then make it non-interactive using the Fiat-Shamir heuristic as a triple,
   $\Pi = (\{CMT^{(k)}\}_{k=1}^t, CH, \{RSP^{(k)}\}_{k=1}^t)$, where
   $\mathrm{CH} = (\{Ch^{(k)}\}_{k=1}^t) = \mathcal{H}_2(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$.
7. Compute $\mathcal{OTS}; sig = \mathsf{OSig}(\mathbf{osk}, (\mathbf{c}_1, \mathbf{c}_2, \Pi))$.
8. Output 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 Σ, the algorithm Verify checks whether the given signature Σ is valid on the given message M and signer is a valid member by executing the following steps.

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

6. For $k = 1$ to t run the verification steps of the commitment scheme 4.1 with public parameter ($\mathbf{A}$, $\mathbf{u}$, $\mathbf{B}$, $\mathbf{V}$, $\mathbf{v}$, $\mathbf{P}$, $\mathbf{c}$) to validate $RSP^{(k)}$ with respect to $CMT^{(k)}$ and $Ch^{(k)}$. If any of the conditions fails then output invalid and hold.
7. For each $\mathbf{u}_i \in RL$ compute $\mathbf{e}_i^{'} = \mathbf{v} - \mathbf{V} \cdot \mathbf{u}_i \mod q$ to check whether there exists an index i such that $||\mathbf{e}_i^{'}||_\infty \leq \beta$. If so return invalid.
8. Return valid.

Open: The algorithm Open($\mathbf{gmsk}$, M, Σ) functions as follow, where $\mathbf{gmsk} = \mathbf{T_B}$ and $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig)$.

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

5. Correctness and Security of the Scheme

5.1. Correctness

Since we use the techniques in [14] and adapt the scheme provided in [10], Verify and Open in our scheme are also correct as the underlying arguments are same. Even though we have changed the revocation token generation regards to [10] there is no impact to the correctness of the scheme from new revocation token generation, since we check the signer's authenticity with RL separately.

For all n, N, all ($\mathbf{gpk}$, $\mathbf{gmsk}$, $\mathbf{gsk}$, $\mathbf{grt}$) outputted by KeyGen(n, N), all $d \in \{0, 1, \dots, N-1\}$, and all $M \in \{0,1\}^*$, Verify($\mathbf{gpk}$, RL, M, Sign($\mathbf{gpk}$, $\mathbf{gsk}[d]$, $\mathbf{grt}[d]$, M)) = valid; $\mathbf{grt}[d] \notin RL$ and Open($\mathbf{gmsk}$, M, Sign($\mathbf{gpk}$, $\mathbf{gsk}[d]$, $\mathbf{grt}[d]$, 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 $\mathbf{V} \in \mathbb{Z}_q^{m \times n}$,*

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

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

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

With overwhelming probability an honest signer (user) can get a valid witness ($\mathbf{x}, \mathbf{r}, \mathbf{e}_1, \mathbf{e}$) to be used in the underlying argument system. Moreover, in the verification algorithm Verify will not return Invalid after the step 6 because steps 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. We can show for this case the verification algorithm outputs Valid with overwhelming probability. According to Lemma 1, $||\mathbf{V} \cdot \mathbf{s}_i||_\infty > 2\beta$ with overwhelming probability. On the other hand, $||\mathbf{V} \cdot \mathbf{s}_i||_\infty \leq ||\mathbf{e}'_i||_\infty + ||\mathbf{e}_1||_\infty \leq ||\mathbf{e}'_i||_\infty + \beta$. Thus, $||\mathbf{e}'_i||_\infty > 2\beta - \beta = \beta$.

Moreover, if the index of the signer is correctly encrypted in the ciphertext $\mathbf{c}$ 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, since 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 $\mathbf{c}$ 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.

5.2. Almost-full Anonymity

Theorem 1. *In the random oracle model, the prosed VLR group signature scheme 4.3 is almost-full anonymous under the $LWE_{n,q,\chi}$ assumption.*

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

Game G_0: This is the real anonymity game, that we assume the adversary A has advantage ϵ. At first the challenger C runs KeyGen(n, N) and generates keys **gpk**, **gmsk**, $\mathbf{gsk}[d]_{d\in\{0,1\}^\ell}$, and $\mathbf{grt}[d]_{d\in\{0,1\}^\ell}$. Then **gpk** and **gsk** are given to the adversary A. The adversary A can query for revocation tokens. When the adversary A requests for a token of user d then the challenger C returns $\mathbf{grt}[d]$. The adversary A can also query for opening of any signature and C answers with Open($\mathbf{gmsk}, M, \Sigma$) using **gmsk** $\mathbf{T_B}$. In the challenge phase A outputs 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 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 task is to determine the index, that is used to generate Σ^*. Thus A outputs $b' \in \{0,1\}$. If $b' = b$ then A wins.

Game G_1: In this game, a slight change is done compared to G_0. The $\mathcal{OTS}$ key pair $(\mathbf{ovk}^*, \mathbf{osk}^*)$ is generated at the beginning of the game. In the real game, $\mathcal{OTS}$ key pair is generated when generating a signature at the challenging phase. Thus, at the query phase if the adversary A queries for opening oracle with a signature $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$, where $\mathbf{ovk} = \mathbf{ovk}^*$ then the challenger C outputs a random bit and aborts. The games G_0 and G_1 are indistinguishable and $\mathbf{ovk}^*$ is independent of the adversary's request as it generated before the query phase. Accordingly, the probability of $\mathbf{ovk} = \mathbf{ovk}^*$ is negligible. Besides, after 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. Hence, the challenger aborting the game is negligible. Without losing the generality assume that A does not request for opening with a valid Σ with $\mathbf{ovk} = \mathbf{ovk}^*$.

Game G_2: In this game, 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 obtain uniformly random $\mathbf{B}^* \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{G}^* \in \mathbb{Z}_q^{n\times \ell}$. To answer the opening oracle requests with $\Sigma = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig, \mathbf{v}, \rho)$ the challenger C samples $\mathbf{Y} \leftarrow (D_{\mathbb{Z}^m,\sigma})^\ell$, computes $\mathbf{G} = \mathbf{B}^*\mathbf{Y} \in \mathbb{Z}_q^{n\times\ell}$, and programs $\mathcal{H}_1(\mathbf{ovk}^*) = \mathbf{G}$. This **G** is used to answer the opening and keep the track of (**ovk**,**Y**,**G**) to be reused if A repeats the same requests 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 generate Σ^*. Since the distributions of $\mathbf{G}^*, \mathbf{B}^*$ are statistically close to the real game [21] and the distribution of $\mathbf{G}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times\ell}$ [21] this game is indistinguishable from the game $\mathbf{G}_1$.

Game G_3: In this game, instead of generating the legitimate non-interactive proof Π, the challenger C simulates Π as discussed in Section 4.1. For each $k \in [t]$ take a fake challenge $\bar{Ch}^{(k)}$ and run the interactive protocol. Then program the random oracle $\mathcal{H}_1$ accordingly. The challenger's signature $\Sigma^* = (\mathbf{ovk}^*, (\mathbf{c}_1^*, \mathbf{c}_2^*), \Pi^*, sig^*, \mathbf{v}^*, \rho^*)$ is statistically close to the signature in the previous games since the argument system is statistically zero-knowledge. Therefore G_3 is indistinguishable from G_2.

Game G_4: In this game, we replace the original 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 χ, and we replace only $\mathbf{grt}[i_b]$ by $\mathbf{t}$. The rest of the game is same as previous game G_3. Thus, the two games are statistically indistinguishable.

Game G_5: In this game we obtain $\mathbf{v}$ uniformly. Thus, we make details of revocation token totally independent of the bit b. We sample $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ and set $\mathbf{v} = \mathbf{y}$. In the previous 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 $\mathbf{y}$. Under the assumption that 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 simply can answer with a value chosen uniformly random such as $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ instead of providing $\mathbf{grt}$. Rest of the game is 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 is succeed. But this contradicts the hardness of the $LWE_{n,q,\chi}$ problem.

Game G_6: In this game we modify the generation 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. The rest of the game is same as G_5. The games G_5 and G_6 are indistinguishable under the assumption of the hardness of $LWE_{n,q,\chi}$ problem defined in Section 2. Indeed, if A can distinguish two games, then A can also solve the LWE problem. That means, he 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 $LWE_{n,q,\chi}$ assumption.

Game G_7: In this game, we make Σ^* 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. Since this game G_7 is independent from the challenger's bit b, the advantage of the adversary winning the game Adv_A is 0.

Even the adversary A can do revocation query any number of times he can not learn about the secret keys since the revocation token consists of part of the public key and some randomness.

Hence, these games prove that advantage of the adversary on almost-full anonymity of the scheme is negligible.

This concludes the proof of anonymity.

5.3. Traceability

We say in the random oracle model our VLR group signture scheme is traceable if the $\mathbf{SIS}^{\infty}_{n,(l+1)m,q,2\beta}$ problem is hard.

Lemma 2 ([21]). *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 ϵ and running time 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 $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$.*

According to Lemma 2 and Theorem 2, we can show that our scheme is traceable in the random oracle under the conditions of those theorems.

Suppose there is an adversary A who can break the computational binding property of the commitment scheme COM with non-negligible probability. Hence A can solve the $\mathbf{SIS}^{\infty}_{n,(\ell+1)m,q,2\beta}$ problem. Thus, without loosing the generality, we assume 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 $(\mathbf{A}, \mathbf{u})$. $\mathcal{F}$ then generates 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 follow.

- **Signatures queries**: If the adversary A queries signature of user d on a random message M, then $\mathcal{F}$ returns $\Sigma = \mathsf{Sign}(\mathbf{gpk}, \mathbf{gsk}[d], \mathbf{grt}[d], M) = (\mathbf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi', sig, \mathbf{v}, \rho)$, where Π' is simulated without using the legitimate secret key and others are generated faithfully. The zeo-knowledge property of the given underlying interactive protocol guarantees that Σ is indistinguishable from the legitimate group signature.
- **Corruption queries**: The corruption set CU is initially set to be empty. If the adversary A queries the secret key of any user d, then $\mathcal{F}$ adds d to the set 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 outputs a message M^*, revocation data RL^* and a non-trivial forged signature Σ^*, which satisfies 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 returns an user index outside of the coalition $CU \setminus RL^*$

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

We require that A always queries $\mathcal{H}_2$ on input $(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$ before $\mathcal{H}_1$. As a result, 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 $(M, \{CMT^{(k)}\}_{k=1}^t, \mathbf{c}_1, \mathbf{c}_2)$. Next, for any fixed k^* run A many times and input as in the original run. For each repeated run, A returns same output for the first k^*-1 queries as in initial run and from the k^*-th query onwards return fresh random values. According to the forking lemma [[29], Lemma 7], with probability larger than $1/2$, algorithm $\mathcal{F}$ can obtain 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 responses of $\mathcal{F}$ with respect to the 3-fork branches be

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

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

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

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

1. $\mathbf{y} = (\mathbf{y}_0 || \mathbf{y}^0_1 || \mathbf{y}^1_1 || \dots || \mathbf{y}^0_\ell || \mathbf{y}^1_\ell) \in \mathsf{Secret}_\beta(d)$ 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 $\mathbf{V} \cdot (\mathbf{B} \cdot \mathbf{r}^*) + \mathbf{e}^*_1 = \mathbf{v} \mod q$.
3. $||\mathbf{e}^*||_\infty \leq b$ and $(\mathbf{B}^T \mathbf{s}^* + \mathbf{e}^*_1 = \mathbf{c}_1 \mod q), (\mathbf{G}^T \mathbf{s}^* + \mathbf{e}^*_2 + \lfloor q/2 \rfloor d^* = \mathbf{c}_2 \mod q)$.

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, $\mathbf{grt}[j^*] \notin RL$, 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 and returns in polynomial time, then so does $\mathcal{F}$.

This concludes the proof of traceability.

6. Conclusion and Open Problems

This paper presented a lattice-based scheme 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 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.

Abbreviations

The following abbreviations are used in this manuscript:

A	Adversary
Adv	Advantage
C	Challenger
DS	Digital Signatures
E	Encryption scheme
FDGS	Fully dynamic group signature
GS	Group Signature
NIZK	Non-interactive zero knowledge
P	Prover
RL	Revocation list
V	Verifier
VLR	Verifier-local revocation

Appendix. Underlying Zero-Knowledge Interactive Protocol

Let n be the security parameter and ℓ be the message length. Let modulus $q = \omega(n^2 \log n)$ be prime, dimension $m \geq 2n \log q$, and Gaussian parameter $\sigma = \omega(\sqrt{n \log q \log n})$. The infinity norm bound $\beta = \lceil \sigma \cdot \log m \rceil$ s.t $(4\beta + 1)^2 \leq q$ and norm bound for LWE noises is b s.t $q/b = \ell \tilde{O}(n)$. Let $k_1 := m + \ell$ and $k_2 := n + m + \ell$.

- The common inputs: Matrices $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}^0_1 | \mathbf{A}^1_1 | \dots | \mathbf{A}^0_\ell | \mathbf{A}^1_\ell] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, $\mathbf{B} \in \mathbb{Z}^{n \times m}$, $\mathbf{V} \in \mathbb{Z}_q^{m \times n}$, and $\mathbf{P} \in \mathbb{Z}_q^{k_1 \times k_2}$ and vectors $\mathbf{u} \xleftarrow{\$} \mathbb{Z}^n_q$, $\mathbf{v} \in \mathbb{Z}^m_q$, and $\mathbf{c} \in \mathbb{Z}^{k_1}_q$.

- 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 $\mathbf{r} \in \mathbb{Z}_q^n$, 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 goal is to convince the verifier in zero-knowledge that:
  - $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \mod q$ and $\mathbf{x} \in \mathsf{Secret}_\beta(d)$.
  - $||\mathbf{f}||_\infty \leq \beta$ and $\mathbf{V} \cdot (\mathbf{B} \cdot \mathbf{r}) + \mathbf{f} = \mathbf{v} \mod q$. (Here the revocation token is created separately with a matrix $\mathbf{B}$ and a vector $\mathbf{r}$ instead of using $\mathbf{A}_0$ and $\mathbf{x}_0$).
  - $||\mathbf{e}||_\infty \leq b$ 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 $\bar{p} = \lfloor \log b \rfloor + 1$).

Before the interaction, both the prover and the verifier form the public matrices: $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$, $\mathbf{V}^* = \mathbf{V} \cdot \mathbf{B} \in \mathbb{Z}_q^{m \times m}$, $\mathbf{I}^* \in \{0,1\}^{m \times 3m}$ ($\mathbf{I}^*$ is obtained 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

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

Then the prover uses the Decomposition-Extension technique provided in [10] with his 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 $\tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_p \leftarrow \mathsf{EleDec}(\mathbf{r})$, then for each $i \in [p]$, let $\mathbf{r}_i \leftarrow \mathsf{EleExt}(\tilde{\mathbf{r}}_i)$.
- Let $\tilde{\mathbf{e}}_1, \ldots, \tilde{\mathbf{e}}_{\bar{p}} \leftarrow \mathsf{EleDec}(\mathbf{e})$, then for each $i \in [p]$, let $\mathbf{e}_i \leftarrow \mathsf{EleExt}(\tilde{\mathbf{e}}_i)$.

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

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

Description of the protocol:

1. **Commitment**: The prover samples randomness ρ_1, ρ_2, ρ_3 for COM and the following uniformly random objects:

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

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

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{r,j}\}_{j=1}^{p}), \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{k}_{z,j}); \\ \quad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{k}_{r,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{k}_{f,j}); \\ \quad \{\pi_{e,j}, \}_{j=1}^{\bar{p}}; \mathbf{P}^* (\sum_{j=1}^{\bar{p}} b_j \mathbf{k}_{e,j}) + \mathbf{Q}\mathbf{k}_d; \tau; \rho_1), \\ \mathbf{c}_2 = \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{k}_{z,j}), \pi_{f,j}(\mathbf{k}_{f,j}), \pi_{r,j}(\mathbf{k}_{r,j})\}_{j=1}^{p}; \\ \quad \{\pi_{e,j}(\mathbf{k}_{e,j})\}_{j=1}^{\bar{p}}; \tau(\mathbf{k}_d); \rho_2), \\ \mathbf{c}_3 = \\ \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{z}_j + \mathbf{k}_{z,j}), \pi_{f,j}(\mathbf{f}_j + \mathbf{k}_{f,j}), \pi_{r,j}(\mathbf{r}_j + \mathbf{k}_{r,j})\}_{j=1}^{p}; \\ \quad \{\pi_{e,j}(\mathbf{e}_j + \mathbf{k}_{e,j})\}_{j=1}^{\bar{p}}; \tau(d^* + \mathbf{k}_d); \rho_3). \end{cases} \tag{A2}$$

864 2. **Challenge:** The verifier sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$ to the prover.

865 3. **Response:** Depending on the challenge, the prover sends the response RSP computed as follows.

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

$$RSP = (d_1, \{\mathbf{u}_{z,j}, \mathbf{w}_{z,j}, \mathbf{u}_{f,j}, \mathbf{w}_{f,j}, \mathbf{u}_{r,j}, \mathbf{w}_{r,j}\}_{j=1}^{p}, \{\mathbf{u}_{e,j}, \mathbf{w}_{e,j}\}_{j=1}^{\bar{p}}, \mathbf{u}_d, \mathbf{w}_d, \rho_2, \rho_3). \tag{A3}$$

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

$$RSP = (d_2, \{\phi_{z,j}, \phi_{f,j}, \phi_{r,j}, \mathbf{s}_{z,j}, \mathbf{s}_{f,j}, \mathbf{s}_{k,j}\}_{j=1}^{p}, \{\phi_{e,j}, \mathbf{s}_{e,j}\}_{j=1}^{\bar{p}}, \hat{\tau}, \mathbf{s}_d, \rho_1, \rho_3) \tag{A4}$$

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

$$RSP = (d_3, \{\psi_{z,j}, \psi_{f,j}, \psi_{r,j}, \mathbf{h}_{z,j}, \mathbf{h}_{f,j}, \mathbf{h}_{k,j}\}_{j=1}^{p}, \{\psi_{e,j}, \mathbf{h}_{e,j}\}_{j=1}^{\bar{p}}, \tilde{\tau}, \mathbf{h}_d, \rho_1, \rho_2) \tag{A5}$$

866 4. Receiving the response RSP, the verifier proceeds as follows:

867
- $Ch = 1$: Parse RSP as in (A3).
  Check whether $\forall \in [p] : \mathbf{u}_{z,j} \in \mathsf{SecretExt}(d_1), \mathbf{u}_{f,j} \in \mathsf{B}_{3m}, \mathbf{u}_{r,j} \in \mathsf{B}_{3m}, \forall j \in [\bar{p}] : \mathbf{u}_d \in \mathsf{B}_{2\ell}, \mathbf{u}_{e,j} \in \mathsf{B}_{3k_2}$, and

$$\begin{cases} \mathbf{c}_2 = \mathsf{COM}(\{\mathbf{w}_{z,j}, \mathbf{w}_{f,j}, \mathbf{w}_{r,j}\}_{j=1}^{p}; \{\mathbf{w}_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{w}_d; \rho_2), \\ \mathbf{c}_3 = \mathsf{COM}(\{\mathbf{u}_{z,j} + \mathbf{w}_{z,j}, \mathbf{u}_{f,j} + \mathbf{w}_{f,j}, \mathbf{u}_{r,j} + \mathbf{w}_{r,j}\}_{j=1}^{p}; \\ \qquad \{\mathbf{u}_{e,j} + \mathbf{w}_{e,j}\}_{j=1}^{\bar{p}}; \{\mathbf{u}_d + \mathbf{w}_d\}; \rho_3). \end{cases} \tag{A6}$$

- $Ch = 2$: Parse RSP as in (A4). Check whether :

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}(d_2, \{\phi_{z,j}, \phi_{f,j}, \phi_{r,j}\}_{j=1}^{p}, \mathbf{A}^* (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{z,j}) - \mathbf{u}; \\ \quad \mathbf{V}^* (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{r,j}) + \mathbf{I}^* (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{f,j}) - \mathbf{v}; \\ \quad \{\phi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{s}_{e,j}) + \mathbf{Q}\mathbf{s}_d - \mathbf{c}; \hat{\tau}; \rho_1), \\ \mathbf{c}_3 = \mathsf{COM}(\{T_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}), \phi_{f,j}(\mathbf{s}_{f,j}), \phi_{r,j}(\mathbf{s}_{r,j})\}_{j=1}^{p}; \\ \quad \{\phi_{e,j}(\mathbf{s}_{e,j})\}_{j=1}^{\bar{p}}; \hat{\tau}(\mathbf{s}_d); \rho_3). \end{cases} \tag{A7}$$

- $Ch = 3$:Parse RSP as in (A5). Check whether :

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}(d_3, \{\psi_{z,j}, \psi_{f,j}, \psi_{r,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{z,j}); \\ \quad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{r,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{f,j}); \\ \quad \{\phi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{h}_{e,j}) + \mathbf{Q}\mathbf{h}_d; \tilde{\tau}; \rho_1), \\ \mathbf{c}_2 = \mathsf{COM}(\{T_{d_3} \circ \psi_{z,j}(\mathbf{h}_{z,j}), \psi_{f,j}(\mathbf{h}_{f,j}), \psi_{r,j}(\mathbf{h}_{r,j})\}_{j=1}^{p}; \\ \quad \{\psi_{e,j}(\mathbf{h}_{e,j}), \}_{j=1}^{\bar{p}}; \tilde{\tau}(\mathbf{h}_d); \rho_2). \end{cases} \tag{A8}$$

The verifier outputs Valid if and only if all the conditions hold. Otherwise, he outputs Invalid.

Appendix A.1. Analysis of the 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 followings.

- There exists an efficient simulator that, on input $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{V}, \mathbf{v}, \mathbf{P}, \mathbf{c})$, 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 3 valid responses $(RSP^{(1)}, RSP^{(2)}, RSP^{(3)})$ corresponding to all 3 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. $||\mathbf{f}'||_\infty \le \beta$ and $\mathbf{V}\cdot(\mathbf{B}\cdot\mathbf{r}) + \mathbf{f}' = \mathbf{v} \mod q$.
  3. $||\mathbf{e}'||_\infty \le b$ and $\mathbf{P}\mathbf{e}' + (0^{k_1-\ell}||\lfloor q/2\rfloor d) = \mathbf{c} \mod q$.

Appendix A.1.1. Completeness and Soundness

An honest prover, with a valid witness $(\mathbf{x}, \mathbf{f}, \mathbf{r}, \mathbf{e})$ for some $d \in \{0,1\}^\ell$, can always obtain $\mathbf{z}_1, \ldots, \mathbf{z}_p \in \mathsf{Secret}_\beta(d), \mathbf{f}_1, \ldots, \mathbf{f}_p \in \mathsf{B}_{3m}, \mathbf{r}_1, \ldots, \mathbf{r}_p \in \mathsf{B}_{3m}$, and $\mathbf{e}_1, \ldots, \mathbf{e}_{\bar{p}} \in \mathsf{B}_{3k_2}$ via the Decomposition-Extension technique [10]. If he follows the protocol, he 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.1.2. Communication Cost

The KTX scheme [27] 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, p permutations in S_{3m}, $\bar{p}$ permutations in S_{3k_2}, one permutation in 2ℓ, p vectors in $\mathbb{Z}_q^{(2\ell+1)3m}$, p 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.1.3. 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**, **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 $\mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}'_j) = \mathbf{u} \mod q$, $\mathbf{r}'_1, \ldots, \mathbf{r}'_p \in \mathbb{Z}_q^{3m}$ and $\mathbf{f}'_1, \ldots, \mathbf{f}'_p \in \mathbb{Z}_q^{3m}$ such that $\mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}'_j) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j) = \mathbf{v} \mod q$, 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 as in equation (A1) and sends commitment $\mathbf{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\hat{\mathcal{V}}$, where

$$\begin{cases} \mathbf{c}'_1 = \mathsf{COM}(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{r,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{k}_{z,j}); \\ \quad \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{k}_{r,j}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{k}_{f,j}) \\ \quad \{\pi_{e,j}\}_{j=1}^{\bar{p}}, \mathbf{P}^* \cdot (\sum_{j=1}^{\bar{p}} b_j \cdot \mathbf{k}_{e,j}) + \mathbf{Q}\mathbf{k}_d; \tau; \rho_1), \\ \mathbf{c}'_2 = \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{k}_{z,j}), \pi_{f,j}(\mathbf{k}_{f,j}), \pi_{r,j}(\mathbf{k}_{r,j})\}_{j=1}^{p}; \\ \quad \{\pi_{e,j}(\mathbf{k}_{e,j})\}_{j=1}^{\bar{p}}; \tau(\mathbf{k}_d); \rho_2), \\ \mathbf{c}'_3 = \mathsf{COM}(\{T_c \circ \pi_{z,j}(\mathbf{z}'_j + \mathbf{k}_{z,j}), \pi_{f,j}(\mathbf{f}'_j + \mathbf{k}_{f,j}), \pi_{r,j}(\mathbf{r}'_j + \mathbf{k}_{r,j})\}_{j=1}^{p}; \\ \quad \{\pi_{e,j}(\mathbf{e}'_j + \mathbf{k}_{e,j})\}_{j=1}^{\bar{p}}; \tau(d' + \mathbf{k}_d); \rho_3). \end{cases} \tag{A9}$$

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

- If Ch = 1: Output $\perp$ and abort.

- If Ch = 2: Send,

RSP = $(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{r,j}, \mathbf{z}'_j + \mathbf{k}_{z,j}, \mathbf{f}'_j + \mathbf{k}_{f,j}, \mathbf{r}'_j + \mathbf{k}_{r,j}\}_{j=1}^{p}$, $\{\pi_{e,j}, \mathbf{e}'_j + \mathbf{k}_{e,j}\}_{j=1}^{\bar{p}}, d' + \mathbf{k}_d, \tau, \rho_1, \rho_3)$.

- If Ch = 3: Send, RSP = $(c, \{\pi_{z,j}, \pi_{f,j}, \pi_{r,j}, \mathbf{k}_{z,j}, \mathbf{k}_{f,j}, \mathbf{k}_{r,j}\}_{j=1}^{p}$, $\{\pi_{e,j}, \mathbf{k}_{e,j}\}_{j=1}^{\bar{p}}, \tau, \rho_1, \rho_2)$.

Case $\overline{Ch} = 2$: *SIM* samples randomness ρ_1, ρ_2, ρ_3 for **COM** and

$$\begin{cases} \hat{d} \xleftarrow{\$} \{0,1\}^{\ell}, c \xleftarrow{\$} \{0,1\}^{\ell}; d' \xleftarrow{\$} \mathsf{B}_{2\ell}; \\ \mathbf{z}'_1, \ldots, \mathbf{z}'_p \xleftarrow{\$} \mathsf{SecretExt}(d); \mathbf{f}'_1, \ldots, \mathbf{f}'_p \xleftarrow{\$} \mathsf{B}_{3m}; \mathbf{r}'_1, \ldots, \mathbf{r}'_p \xleftarrow{\$} \mathsf{B}_{3m}; \\ \quad \mathbf{e}'_1, \ldots, \mathbf{e}'_{\bar{p}} \xleftarrow{\$} \mathsf{B}_{3k}; \\ \pi_{z,1}, \ldots, \pi_{z,p} \xleftarrow{\$} \mathcal{S}; \pi_{f,1}, \ldots, \pi_{f,p} \xleftarrow{\$} \mathcal{S}_{3m}; \pi_{r,1}, \ldots, \pi_{r,p} \xleftarrow{\$} \mathcal{S}_{3m}; \\ \quad \pi_{e,1}, \ldots, \pi_{e,\bar{p}} \xleftarrow{\$} \mathcal{S}_{3k}; \\ \mathbf{k}_{z,1}, \ldots, \mathbf{k}_{z,p} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)3m}; \mathbf{k}_{f,1}, \ldots, \mathbf{k}_{f,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{k}_{r,1}, \ldots, \mathbf{k}_{r,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}; \\ \quad \mathbf{k}_{e,1}, \ldots, \mathbf{k}_{e,\bar{p}} \xleftarrow{\$} \mathbb{Z}_q^{3k}; \mathbf{k}_d \xleftarrow{\$} \mathbb{Z}_q^{2\ell}, \tau \xleftarrow{\$} \mathcal{S}_{2\ell}. \end{cases}$$

Next *SIM* forms and sends commitment CMT as the same manner as in (A9).

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

- If Ch = 1: $(\hat{d} \oplus c\ \{T_c \circ \pi_{z,j}(\mathbf{z}'_j), T_c \circ \pi_{z,j}(\mathbf{k}_{z,j}), \pi_{f,j}(\mathbf{f}'_j), \pi_{f,j}(\mathbf{k}_{f,j}),$ $\pi_{r,j}(\mathbf{r}'_j), \pi_{r,j}(\mathbf{k}_{r,j})\}_{j=1}^{p}, \{\pi_{e,j}(\mathbf{e}'_j), \pi_{e,j}(\mathbf{k}_{e,j})\}_{j=1}^{\bar{p}}, \tau(d'), \tau(\mathbf{k}_d))$.

- 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 (A9), and

$$\begin{aligned}\mathbf{c}_1' = \mathsf{COM}\ (c, \{\pi_{z,j}, \pi_{e,j}, \pi_{r,j}\}_{j=1}^{p}, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{z}_j' + \mathbf{k}_{z,j})) - \mathbf{u};\\ \mathbf{V}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{r}_j' + \mathbf{k}_{r,j})) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{f}_j' + \mathbf{k}_{f,j})) - \mathbf{v};\\ \{\pi_{e,j}\}_{j=1}^{\bar{p}}; \mathbf{P}^* \sum_{j=1}^{\bar{p}} b_j(\mathbf{e}_j' + \mathbf{k}_{e,j}) + \mathbf{Q}(d' + \mathbf{k}_d) - \mathbf{c}; \tau; \rho_1).\end{aligned}$$

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.

Since **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. Hence, 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.1.4. 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.

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

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

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

For each $j \in [p]$, let $\mathbf{y}'_j = (\mathbf{s}_{z,j} - \mathbf{h}_{z,j})$. Then $T_{d_2} \circ \phi_{z,j}(\mathbf{y}'_j) = T_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}) - T_{d_2} \circ \phi_{z,j}(\mathbf{h}_{z,j}) = \mathbf{u}_{z,j} \in \mathsf{SecretExt}(d_1)$. 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 $j \in [p]$, $\mathbf{y}'_j \in \mathsf{SecretExt}(\bar{d})$, since 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

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

Moreover, since $\mathbf{y}'_j \in \mathsf{SecretExt}(\bar{d})$ 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 $\phi_{f,j}(\mathbf{f}'_j) = \phi_{f,j}(\mathbf{s}_{f,j}) - \phi_{e,j}(\mathbf{h}_{f,j}) = \mathbf{u}_{f,j} \in \mathsf{B}_{3m}$, 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,

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

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

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

We can obtain the relation:

$$\mathbf{V}^* \cdot \hat{\mathbf{r}} + \mathbf{I}^* \cdot \hat{\mathbf{f}} = \mathbf{v} \mod q \iff \mathbf{V}^* \cdot (\mathbf{B} \cdot \mathbf{r}') + \mathbf{f}' = \mathbf{v} \mod q.$$

Let $d^* = \mathbf{s}_d - \mathbf{h}_d = \hat{\tau}^{-1}(\mathbf{u}_d)$. 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 $\phi_{e,j}(\mathbf{e}'_j) = \phi_{e,j}(\mathbf{s}_{e,j}) - \phi_{e,j}(\mathbf{h}_{e,j}) = \mathbf{u}_{e,j} \in \mathsf{B}_{3k}$, 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,

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

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

References

1. Chaum, D.; Van Heyst, E. Group signatures. EUROCRYPT 1991, LNCS. Springer Berlin Heidelberg, 1991, Vol. 547, pp. 257–265.
2. Chen, L.; Pedersen, T.P. New group signature schemes. Workshop on the Theory and Application of of Cryptographic Techniques, LNCS. Springer Berlin Heidelberg, 1994, Vol. 950, pp. 171–181.
3. Ateniese, G.; Tsudik, G. Group signatures á la carte. Symposium on Discrete Algorithms: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms. SIAM, 1999, Vol. 17, pp. 848–849.
4. Ateniese, G.; Camenisch, J.; Joye, M.; Tsudik, G. A practical and provably secure coalition-resistant group signature scheme. CRYPTO 2000, LNCS. Springer Berlin Heidelberg, 2000, Vol. 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. EUROCRYPT 2003, LNCS. Springer Berlin Heidelberg, 2003, Vol. 2656, pp. 614–629.
6. Bresson, E.; Stern, J. Efficient revocation in group signatures. PKC 2001, LNCS. Springer Berlin Heidelberg, 2001, Vol. 1992, pp. 190–206.
7. Camenisch, J.; Lysyanskaya, A. Dynamic accumulators and application to efficient revocation of anonymous credentials. CRYPTO 2002, LNCS. Springer Berlin Heidelberg, 2002, Vol. 2442, pp. 61–76.
8. Brickell, E. An Efficient Protocol for Anonymously Providing Assurance of the Container of the Private Key. *Submitted to the Trusted Comp. Group (April 2003)* **2003**.
9. Boneh, D.; Shacham, H. Group signatures with verifier-local revocation. ACM-CCS 2004. ACM, 2004, pp. 168–177.
10. Langlois, A.; Ling, S.; Nguyen, K.; Wang, H. Lattice-Based Group Signature Scheme with Verifier-Local Revocation. PKC 2014, LNCS. Springer Berlin Heidelberg, 2014, Vol. 8383, pp. 345–361.
11. Gordon, S.D.; Katz, J.; Vaikuntanathan, V. A group signature scheme from lattice assumptions. ASIACRYPT 2010, LNCS. Springer Berlin Heidelberg, 2010, Vol. 6477, pp. 395–412.
12. Camenisch, J.; Neven, G.; Rückert, M. Fully Anonymous Attribute Tokens from Lattices. SCN 2012, LNCS. Springer Berlin Heidelberg, 2012, Vol. 12, pp. 57–75.
13. Laguillaumie, F.; Langlois, A.; Libert, B.; Stehlé, D. Lattice-based group signatures with logarithmic signature size. ASIACRYPT 2013, LNCS. Springer Berlin Heidelberg, 2013, Vol. 8270, pp. 41–61.
14. Ling, S.; Nguyen, K.; Wang, H. Group signatures from lattices: simpler, tighter, shorter, ring-based. PKC 2015, LNCS. Springer Berlin Heidelberg, 2015, Vol. 9020, pp. 427–449.
15. Boyen, X. Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. PKC 2010, LNCS. Springer Berlin Heidelberg, 2010, Vol. 6056, pp. 499–517.
16. Nguyen, P.Q.; Zhang, J.; Zhang, Z. Simpler efficient group signatures from lattices. PKC 2015, LNCS. Springer Berlin Heidelberg, 2015, Vol. 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. ASIACRYPT 2016, LNCS. Springer Berlin Heidelberg, 2016, Vol. 10032, pp. 373–403.
18. Perera, M.N.S.; Koshiba, T. Fully Dynamic Group Signature Scheme with Member Registration and Verifier-local Revocation. ICMC 2018, Mathematics and Computing, to appear.
19. Peikert, C. A Decade of Lattice Cryptography. *Foundations and Trends in Theoretical Computer Science* **2016**, *10*, 283–424.
20. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. STOC 2005. ACM Press, 2005, pp. 84–93.
21. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. ACM 2008. ACM, 2008, pp. 197–206.
22. Ajtai, M. Generating hard instances of lattice problems. Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. ACM, 1996, pp. 99–108.
23. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. EUROCRYPT 2012. Springer Berlin Heidelberg, 2012, Vol. 7237, pp. 700–718.

24. Agrawal, S.; Boyen, X.; Vaikuntanathan, V.; Voulgaris, P.; Wee, H. Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices. PKC 2012, LNCS. Springer Berlin Heidelberg, 2012, Vol. 7293, pp. 280–297.
25. Naor, D.; Shenhav, A.; Wool, A. One-time signatures revisited: Have they become practical? *IACR Cryptology ePrint Archive* **2005**, *2005*, 442.
26. Chow, S.S.; Wong, D.S. Anonymous identification and designated-verifiers signatures from insecure batch verification. EuroPKI 2007, LNCS. Springer Berlin Heidelberg, 2007, Vol. 4582, pp. 203–219.
27. Kawachi, A.; Tanaka, K.; Xagawa, K. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. ASIACRYPT 2008, LNCS. Springer Berlin Heidelberg, 2008, Vol. 5350, pp. 372–389.
28. Perera, M.N.S.; Koshiba, T. Zero-knowledge proof for Lattice-based group signature schemes with Verifier-local Revocation. 9-th International Workshop on Trustworthy Computing and Security (TwCSec-2018), LNDT, to appear.
29. Pointcheval, D.; Vaudenay, S. *On Provable Security for Digital Signature Algorithms*; Ecole Normale Supérieure (Paris). Laboratoire d'Informatique, 1996.