# A Lattice-Based Group Signature Scheme with Verifier-Local Revocation

San Ling[1], Khoa Nguyen[1], Adeline Roux-Langlois[2], and Huaxiong Wang[1]

[1] Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
[2] CNRS/IRISA, 263 avenue du Général Leclerc, 35042 Rennes, France

**Abstract.** Support of membership revocation is a desirable functionality for any group signature scheme. Among the known revocation approaches, verifier-local revocation (VLR) seems to be the most flexible one, because it only requires the verifiers to possess some up-to-date revocation information, but not the signers. All of the contemporary VLR group signatures operate in the bilinear map setting, and all of them will be insecure once quantum computers become a reality. In this work, we introduce the first lattice-based VLR group signature, and thus, the first such scheme that is believed to be quantum-resistant. In comparison with existing lattice-based group signatures, our scheme has several noticeable advantages: support of membership revocation, logarithmic-size signatures, and milder hardness assumptions. Moreover, our construction works without relying on public-key encryption schemes, which is an intriguing feature for group signatures.

## 1 Introduction

Group signatures have been an important research topic in public-key cryptography since their introduction by Chaum and van Heyst [15]. In these schemes, all the potential signers form a group, where each signer can anonymously issue a signature on behalf of the whole group (anonymity). On the other hand, in cases of disputes, there is a tracing mechanism which can link a given signature to the identity of the misbehaving member (traceability). These two attractive features allow group signatures to find applications in various real-life scenarios, such as anonymous online communications, digital right management, e-commerce systems, and much more. Over the last two decades, many group signature schemes with different security models, different levels of efficiency and functionality have been proposed ([16,4,5,8,9,6,21,28], ...).

One desirable functionality of group signatures is the support for membership revocation. For example, misbehaving members who issue signatures for documents, which they are not allowed to sign, should be revoked from the group. In these cases, if a group signature scheme does not support revocation, then the whole system has to be re-initialized, which is obviously an unsuitable solution in practice. Currently there are two main revocation approaches for group signatures. The first approach requires all the unrevoked members to update their signing keys after each revocation ([4,12,8,11],...). At the same time, all the signature verifiers need to download the up-to-date group public key. As a consequence, it is sometimes inconvenient to practically implement such schemes. The second approach, that is group signatures with verifier-local revocation (VLR), only requires the verifiers to possess some up-to-date revocation information, but not the signers. Since in most of real-life scenarios, the number of signature verifiers is much smaller than the number of signers, this revocation approach is more flexible and more practical. Moreover, it is akin to that of the traditional Public Key Infrastructures, where the verifiers use the latest Certificate Revocation List to check the public key of the signer. The notion of VLR group signatures was introduced by Brickell [10], then formalized by Boneh and Shacham [9], further investigated and extended by Nakanishi and Funabiki [39,40], Libert and Vergnaud [29], and Bichsel et al. [7]. It is worth mentioning that all the existing VLR group signatures scheme operate in the bilinear map setting. Furthermore, all these schemes will be insecure once quantum computers become a reality [47]. Thus, constructing a VLR group signature schemes which is secure against quantum computers, or even outside of the bilinear map setting, is a challenging open question.

Lattice-based group signatures. Lattice-based cryptography is currently considered as the most promising candidate for post-quantum cryptography. As opposed to classical cryptography (i.e., based on the hardness of factoring or discrete log problems), lattice-based cryptography is widely believed to be resistant against quantum computers, moreover, it enjoys provable security under *worst-case* hardness assumptions ([1,45,19,37]). Designing secure and efficient lattice-based cryptographic constructions (and group signatures, in particular) becomes an intriguing challenge for the research community looking forward to the future. To the best of our knowledge, three lattice-based group signature schemes have been proposed, but none of them supports membership revocation. The first one was introduced by Gordon et al. [20] in 2010. While their scheme is of great theoretical interest, its signatures have size $\mathcal{O}(N)$, where $N$ is the number of group users. In terms of efficiency, this is a noticeable disadvantage if the group is large, e.g., group of all employees of a big company. Camenisch et al. [13] later proposed lattice-based anonymous attribute tokens system, a primitive that can be considered as a generalization of group signature. However, in their construction, the signatures size is still linear in $N$. Recently, Laguillaumie et al. [23] designed a scheme featuring signature size $\widetilde{\mathcal{O}}(\log N)$, which is the first lattice-based group signature that overcomes the linear-size barrier. We remark that all the above mentioned schemes follow the traditional sign-and-encrypt-and-prove paradigm: to enable the tracing mechanism, these schemes require the signer to encrypt some private information via certain type of public-key encryption (PKE) based on the Learning With Errors ($\mathsf{LWE}$) problem, and then generate a sophisticated proof to prove particularly that the ciphertext is well-formed. Relying on PKE to construct group signatures may imply two troublesome issues: firstly, it makes the construction less efficient; secondly, since the whole system is secure only if the underlying PKE scheme is secure, it sometimes does lead to a relatively strong hardness assumption. In particular, the recent scheme by Laguillaumie et al. [23] is only provably secure if there is no quantum algorithm to approximate the Shortest Independent Vectors Problem ($\mathsf{SIVP}_\gamma$) on lattices of dimension $n$ to within certain $\gamma = \widetilde{\mathcal{O}}(n^{8.5})$. This yields several interesting open questions in this direction: Is it possible to construct a scheme that supports membership revocation? Can lattice-based group signature schemes be free of LWE-based PKE? How to design a more efficient scheme based on weaker security assumptions?

Our contributions. In the present work, we reply to all the above open questions positively. In particular, we introduce the first group signature with verifier-local revocation from lattice assumptions, and thus, the first such scheme that is believed to be quantum-resistant. In comparison with known lattice-based group signatures, while the schemes from [20], [13] and [23] follow the CPA-*anonymity* and CCA-*anonymity* notions from [8,5], our construction satisfies the (weaker) notion of *selfless-anonymity* for VLR group signatures from [9]. Nevertheless, our scheme has several remarkable advantages over the contemporary counterparts:

1. Functionality: Our scheme is the first lattice-based group signature that supports membership revocation. As discussed above, this is a desirable functionality for any group signature scheme.
2. Simplicity: Our scheme is conceptually very simple. Each group signature roughly consists of an LWE instance and an all-in-one argument of knowledge, made non-interactive using Fiat-Shamir paradigm [18]. Moreover, the scheme departs from the traditional paradigm, and is free of LWE-based public-key encryptions.
3. Efficiency: For a security parameter $n$ and for a group of $N$ members, the group public key and the signature have bit-sizes $\widetilde{\mathcal{O}}(n^2) \cdot \log N$ and $\widetilde{\mathcal{O}}(n) \cdot \log N$, respectively. This level of asymptotic efficiency is comparable to that of [23], and is a noticeable improvement over those of [20] and [13].
4. Security assumption: Our scheme is proved to be secure (in the random oracle model) based on the worst-case hardness of approximating the Shortest Independent Vectors Problem, for general lattices of dimension $n$, to within a factor $\gamma = \widetilde{\mathcal{O}}(n^{2.5})$. That is, in contrast to [23], we achieve logarithmic-size signatures without having to pay a heavy cost in terms of hardness assumptions.

Overview of our techniques. The main building block of our VLR group signature scheme is a Stern-like [48] interactive argument system allowing a signer to convince the verifier in zero-knowledge that: (i) the signer is a certified group member (i.e., he possesses a valid secret signing key); (ii) the "revocation token" assigned to the signer is correctly committed via a function that is injective, one-way and pseudo-random. The argument system is repeated many times to make the soundness error negligibly small, and then is converted to a signature scheme via Fiat-Shamir heuristic [18]. If the produced signature is generated by an honest signer whose revocation token is unavailable to the verifier, then it should be accepted and anonymous. On the other hand, if the signer has been revoked, then his token is known to the verifier, which allows the latter to check that the token yields the committed function value and to reject the signature accordingly.

We consider a group of $N = 2^\ell$ users, where each user is identified by a string $d \in \{0,1\}^\ell$ denoting the binary representation of his index in the group. Let $n, m, \beta$, and $q \geq 2$ be integers (to be determined later). Our scheme operates within the structure of a *Bonsai tree* of hard random lattices [14], specified by a matrix $\mathbf{A} = \big[\mathbf{A}_0\big|\mathbf{A}_1^0\big|\mathbf{A}_1^1\big|\ldots\big|\mathbf{A}_\ell^0\big|\mathbf{A}_\ell^1\big] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$, and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. Initially, the group user with identity $d = d[1]\ldots d[\ell] \in \{0,1\}^\ell$ is issued a Bonsai signature on his identity, that is a small vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$, such that $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{A}_d \cdot \mathbf{z} = \mathbf{u} \bmod q$, where $\mathbf{A}_d = \big[\mathbf{A}_0\big|\mathbf{A}_1^{d[1]}\big|\ldots\big|\mathbf{A}_\ell^{d[\ell]}\big]$ - a subtree defined by $d$. In other words, $\mathbf{z}$ is a solution to the Inhomogeneous Small Integer Solution ($\mathsf{ISIS}$) instance $(\mathbf{A}_d, \mathbf{u})$. To prove that he is a certified group member without leaking $\mathbf{z}$, the user can perform a proof/argument of knowledge (e.g., [38,33,30]) to convince the verifier that he knows such a vector $\mathbf{z}$ in zero-knowledge.

At this stage, one can obtain a secure identity-based identification scheme (as shown in [46]), but it is insufficient for our purposes: to achieve anonymity, the group user additionally has to *hide* his identity $d$, and hence, the matrix $\mathbf{A}_d$ should not be explicitly given. This raises an interesting question: If the verifier does not know $\mathbf{A}_d$, how could he be convinced that $\mathbf{A}_d \cdot \mathbf{z} = \mathbf{u} \bmod q$? To address this issue, we introduce the following extension: we add $\ell$ suitable *zero-blocks* of size $m$ to vector $\mathbf{z}$ to obtain the extended vector $\mathbf{x} = \big(\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$, where the added zero-blocks are $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$. We then have $\|\mathbf{x}\|_\infty \leq \beta$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$. Namely $\mathbf{x}$ is a solution to the $\mathsf{ISIS}$ instance given by the *whole* Bonsai tree, with an additional condition: for each $i = 1, \ldots, \ell$, one of the two blocks $\mathbf{x}_i^0, \mathbf{x}_i^1$ must be zero, where the arrangement of the zero-blocks is determined by $d$. To prove in zero-knowledge the possession of such a vector $\mathbf{x}$, we adapt the 'Stern Extension' argument system from [30], where the user identity $d$ is hidden by a *"one-time pad"* technique. This technique is as follows. In each round of the protocol, the user samples a fresh uniformly random $e \in \{0,1\}^\ell$ and permutes the blocks of $\mathbf{x}$ to obtain the permuted vector $\mathbf{v}$, whose zero-blocks are arranged according to $d \oplus e$ (where $\oplus$ denotes the bit XOR operation). Depending on the verifier's challenge, the user later will either reveal $e$, or reveal $d \oplus e$ and show that $\mathbf{v}$ has the correct shape determined by $d \oplus e$. Since $d \oplus e$ is uniformly random over $\{0,1\}^\ell$, the user identity $d$ is completely hidden. As a result, the user can anonymously prove his group membership.

We now sketch our revocation mechanism. For each group user's secret key $\mathbf{x}$, consider the first block $\mathbf{x}_0$ that corresponds to the "root" $\mathbf{A}_0$ of the Bonsai tree, and let his revocation token be $\mathsf{token} = \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q \in \mathbb{Z}_q^n$. We choose suitable parameters, and sample $\mathbf{x}_0$ from a proper distribution, so that the token is statistically close to uniform over $\mathbb{Z}_q^n$. When issuing a VLR group signature, the user draws a uniformly random matrix $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$ from a random oracle, samples a small vector $\mathbf{e} \in \mathbb{Z}^m$ from the $\mathsf{LWE}$ error distribution, outputs the

$\mathsf{LWE}$ function:

$$\mathbf{b} = \mathbf{B} \cdot \mathsf{token} + \mathbf{e} \bmod q,$$

and extends the zero-knowledge argument system discussed above to additionally prove that vector $\mathbf{b}$ is honestly generated. Given a signature containing vector $\mathbf{b}$, and a list of revocation tokens $RL = \{\{\mathbf{u}_i\}_i\}$, the verifier proceeds as follows. First, compute vector $\mathbf{e}'_i = \mathbf{b} - \mathbf{B} \cdot \mathbf{u}_i \bmod q$, for all $i$, and then, reject the signature if there exists index $i$ such that $\|\mathbf{e}'_i\|_\infty$ is small. The soundness of the underlying argument system ensures that a revoked signer cannot pass the test. On the other hand, since the $\mathsf{LWE}$ function is injective, one-way and pseudorandom (for appropriate setting of parameters), signatures issued by non-revoked members should remain anonymous and should not be falsely rejected.

Putting everything together, we obtain a lattice-based VLR group signature that has several nice features, as mentioned earlier. In the process, we exploit the rich structure of the Bonsai tree [14], and the versatility of the "Stern Extension" argument system from [30]. We also employ a special *"one-time pad"* technique for hiding secret bits, and an interesting revocation mechanism.

Remarks. This article is the full, corrected version of [24], published in the proceedings of PKC 2014. As pointed out to us by Shota Yamada, [24] contained a flaw in the revocation mechanism, where we employed a kind of inequality test to check whether the signer's token does not belong to a given list of revocation tokens. That inequality test was not properly designed, and was vulnerable against a traceability adversary. Fortunately, the mistake has been completely fixed in the present article, where we instead realize the revocation mechanism by binding the user's revocation token to an $\mathsf{LWE}$ instance.

Subsequent to the publication of [24], a rich line of work [31,26,27,25,32] has developed Stern-like protocols to design new lattice-based group signatures, providing various improvements in terms of security, efficiency and functionality. Cheng et al. [17] relied on our protocol for the Bonsai tree signature to build a lattice-based policy-based signature scheme.

## 2 Preliminaries

Notations. For a positive integer $n$, we let $[n]$ denote the set $\{1, \ldots, n\}$. Vectors will be denoted in bold lower-case letters and matrices will be denoted in bold upper-case letters. We assume that all vectors are column vectors. The concatenation of vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^k$ is denoted by $(\mathbf{x}\|\mathbf{y})$. We denote the column concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times k}$ by $\big[\mathbf{A}\big|\mathbf{B}\big]$. Let $\mathbf{x} = (x_1, \ldots, x_n)$, we denote by $\mathsf{Parse}(\mathbf{x}, i_1, i_2)$ the vector $(x_{i_1}, x_{i_1+1} \ldots, x_{i_2})$ for $1 \leq i_1 \leq i_2 \leq n$. If S is a finite set, $y \xleftarrow{\$} \mathrm{S}$ means that $y$ is chosen uniformly at random from S. If $D_1$ and $D_2$ are two distributions over the same countable support $S$, then their statistical distance is defined as $\Delta(D_1, D_2) = \frac{1}{2}\sum_{x\in S} |D_1(x) - D_2(x)|$. Two distributions are statistically close if their statistical distance is negligible. The acronym "PPT" stands for "probabilistic polynomial-time."

### 2.1 VLR Group Signature

The presentation in this Section follows [9]. A VLR group signature consists of 3 following algorithms:

- $\mathsf{KeyGen}(n, N)$: This PPT algorithm takes as input security parameter $n$ and group size $N$, and outputs a group public key $\mathsf{gpk}$, a vector of user secret keys $\mathsf{gsk} = (\mathsf{gsk}[0], \mathsf{gsk}[1], \ldots, \mathsf{gsk}[N-1])$, and a vector of user revocation tokens $\mathsf{grt} = (\mathsf{grt}[0], \mathsf{grt}[1], \ldots, \mathsf{grt}[N-1])$.
- $\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$: On input group pubic key $\mathsf{gpk}$, a user secret key $\mathsf{gsk}[d]$, and a message $M \in \{0,1\}^*$, this PPT algorithm outputs a signature $\Sigma$.
- $\mathsf{Verify}(\mathsf{gpk}, RL, \Sigma, M)$: On input group public key $\mathsf{gpk}$, a set of revocation tokens $RL \subseteq \{\mathsf{grt}[0], \mathsf{grt}[1], \ldots, \mathsf{grt}[N-1]\}$, a signature $\Sigma$, and the message $M$, this algorithm outputs either $\mathsf{Valid}$ or $\mathsf{Invalid}$. The output $\mathsf{Valid}$ indicates that $\Sigma$ is a valid signature on message $M$ under $\mathsf{gpk}$, and the signer has not been revoked.

*Remark 1.* Any VLR group signature has an *implicit tracing algorithm* using vector $\mathsf{grt}$ as the tracing key. The tracing algorithm works as follows: on input a valid signature $\Sigma$ on a message $M$, it reveals the signer of $\Sigma$ by running $\mathsf{Verify}(\mathsf{gpk}, RL = \mathsf{grt}[d], \Sigma, M)$, for $d = 0, 1, \ldots$, and outputting the first index $d^* \in \{0, 1, \ldots, N-1\}$ for which the verification algorithm returns $\mathsf{Invalid}$. The tracing algorithm fails if and only if the given signature is properly verified for all $d$.

A secure VLR group signature scheme must satisfy the following 3 requirements:

1. **Correctness**: For all $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt}) \leftarrow \mathsf{KeyGen}(n, N)$, all $d \in \{0, 1, \ldots, N-1\}$, and all $M \in \{0,1\}^*$,

$$\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid} \Leftrightarrow \mathsf{grt}[d] \notin RL.$$

2. **Selfless-anonymity**: In the following selfless-anonymity game, the adversary's goal is to determine which of the two adaptively chosen keys generated a signature. He is not given access to either key.
   (a) Setup. The challenger runs $\mathsf{KeyGen}$ to generate $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$, then gives $\mathsf{gpk}$ to the adversary $\mathcal{A}$.
   (b) Queries. Adversary $\mathcal{A}$ can make the following queries:
   - Signing: Query for signature of any user $d$ on any message $M \in \{0,1\}^*$. The challenger returns the signature $\Sigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$.
   - Corruption: Query for the secret key of any user $d$. The challenger returns $\mathsf{gsk}[d]$.
   - Revocation: Query for the revocation token of any user $d$. The challenger returns $\mathsf{grt}[d]$.

   (c) Challenge. Adversary $\mathcal{A}$ outputs a message $M^*$ and two indices $d_0$ and $d_1$, such that $\mathcal{A}$ never made a corruption or revocation query for user $d_0$ or user $d_1$. The challenger chooses a bit $c \xleftarrow{\$} \{0,1\}$, computes a signature of user $d_c$ on $M^*$ as $\Sigma^* = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d_c], M^*)$, and returns $\Sigma^*$ to $\mathcal{A}$.

(d) $\mathsf{Restricted\ Queries}$. After the challenge phase, $\mathcal{A}$ can still make queries as before, but with the following restrictions: it is not allowed to make any corruption or revocation query for user $d_0$ or user $d_1$.

(e) $\mathsf{Output}$. Eventually, $\mathcal{A}$ outputs a bit $c'$. It wins the game if $c' = c$.

We define the adversary's advantage as $\mathsf{Adv}_{\mathcal{A}} = \big|\Pr[c' = c] - 1/2\big|$. We say that the VLR group signature is selfless-anonymous if $\mathsf{Adv}_{\mathcal{A}}$ is negligible.

3. **Traceability**: The adversary's goal in the traceability game is to forge a signature that cannot be traced to one of the users in his coalition using the implicit tracing algorithm above. The traceability game is defined as follows:

(a) $\mathsf{Setup}$: Run $\mathsf{KeyGen}(n, N)$ to obtain $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$. Adversary $\mathcal{A}$ is given $(\mathsf{gpk}, \mathsf{grt})$. Set $U = \emptyset$.

(b) $\mathsf{Queries}$: Adversary $\mathcal{A}$ can make queries to the following oracles:

- Signing: On input a message $M$, and an index $d$, the oracle returns $\varSigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$.
- Corruption: On input an index $d$, the oracle adds $d$ to the set $U$, and returns $\mathsf{gsk}[d]$.

(c) $\mathsf{Forgery}$: Eventually, $\mathcal{A}$ outputs a message $M^*$, a set of revocation tokens $RL^*$ and a signature $\varSigma^*$.

The adversary wins the game if:

i. $\mathsf{Verify}(\mathsf{gpk}, RL^*, \varSigma^*, M^*) = \mathsf{valid}$.

ii. The (implicit) tracing algorithm fails or traces to a user outside of the coalition $U \setminus RL^*$.

iii. The signature $\varSigma^*$ is non-trivial, i.e., $\mathcal{A}$ did not obtain $\varSigma^*$ by making a signing query on $M^*$.

The probability that $\mathcal{A}$ wins the game, denoted by $\mathsf{SuccPT}_{\mathcal{A}}$, is taken over the randomness of $\mathcal{A}$, algorithms $\mathsf{KeyGen}$ and $\mathsf{Sign}$. We say that a VLR group signature is traceable if $\mathsf{SuccPT}_{\mathcal{A}}$ is negligible.

### 2.2 Some Cryptographic Tools from Lattices

**Lattices.** Let $n, m$, and $q \geq 2$ be integers. For matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define the $m$-dimensional lattice:

$$\varLambda^{\perp}(\mathbf{A}) = \big\{\mathbf{x} \in \mathbb{Z}^m : \;\; \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q\big\} \subseteq \mathbb{Z}^m.$$

For any $\mathbf{u}$ in the image of $\mathbf{A}$, define the coset

$$\varLambda^{\perp}_{\mathbf{u}}(\mathbf{A}) = \big\{\mathbf{x} \in \mathbb{Z}^m : \;\; \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q\big\}.$$

We recall the homogeneous and inhomogeneous Small Integer Solution problems ($\mathsf{SIS}$ and $\mathsf{ISIS}$), the Learning With Errors $\mathsf{LWE}$ problem, as well as their hardness results..

**Definition 1.** *The* $\mathsf{SIS}^p_{n,m,q,\beta}$ *and* $\mathsf{ISIS}^p_{n,m,q,\beta}$ *problems in the* $\ell_p$ *norm with parameters* $(n, m, q, \beta)$ *are as follows. Given a uniformly random matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, and a uniformly random vector* $\mathbf{u} \in \mathbb{Z}_q^n$*,*

- $\mathsf{SIS}^p_{n,m,q,\beta}$ *asks to find a non-zero vector* $\mathbf{x} \in \varLambda^{\perp}(\mathbf{A})$ *such that* $\|\mathbf{x}\|_p \leq \beta$*.*

- $\mathsf{ISIS}^p_{n,m,q,\beta}$ *asks to find a vector* $\mathbf{x} \in \Lambda^{\perp}_{\mathbf{u}}(\mathbf{A})$ *such that* $\|\mathbf{x}\|_p \leq \beta$.

The hardness of the $\mathsf{SIS}$ and $\mathsf{ISIS}$ problems is given by a worst-case to average-case reduction from standard lattice problems, such as the Shortest Independent Vectors Problem ($\mathsf{SIVP}$).

**Lemma 1 ([19]).** *For any* $m$*,* $\beta = \mathrm{poly}(n)$*, and for any* $q \geq \beta \cdot \omega(\sqrt{n \log n})$*, solving a* random *instance of the* $\mathsf{SIS}^2_{n,m,q,\beta}$ *or* $\mathsf{ISIS}^2_{n,m,q,\beta}$ *problem with non-negligible probability is at least as hard as approximating the* $\mathsf{SIVP}_\gamma$ *problem on* any *lattice of dimension* $n$ *to within certain* $\gamma = \beta \cdot \widetilde{O}(\sqrt{n})$ *factors.*

It then follows from the relationship between the $\ell_2$ and $\ell_\infty$ norms that the $\mathsf{SIS}^\infty_{n,m,q,\beta}$ and $\mathsf{ISIS}^\infty_{n,m,q,\beta}$ problems are at least as hard as $\mathsf{SIVP}_\gamma$ (in the $\ell_2$ norm) for some $\gamma = \beta \cdot \widetilde{O}(\sqrt{nm})$.

**Definition 2 ([45]).** Let $n, m \geq 1$, $q \geq 2$, and let $\chi$ be a probability distribution on $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \hookleftarrow \chi$, and outputting the pair $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The $\mathsf{LWE}_{n,q,\chi}$ problem asks to distinguish $m$ samples chosen according to $\mathcal{A}_{\mathbf{s},\chi}$ (for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$) and $m$ samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

If $q$ is a prime power, $\beta \geq \sqrt{n}\omega(\log n)$, $\gamma = \widetilde{\mathcal{O}}(nq/\beta)$, then there exists an efficient sampleable $\beta$-bounded distribution $\chi$ (i.e., $\chi$ outputs samples with norm at most $\beta$ with overwhelming probability) such that the $\mathsf{LWE}_{n,q,\chi}$ problem is as least as hard as $\mathsf{SIVP}_\gamma$ (see [45,41,34,35]).

**Gaussians over Lattices.** For any positive real $\sigma$, the $n$-dimensional Gaussian function is defined as: $\forall \mathbf{x} \in \mathbb{R}^n$, $\rho_\sigma(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2/\sigma^2)$. For any $n$-dimensional lattice $\Lambda$, define the discrete Gaussian distribution over $\Lambda$ as: $\forall \mathbf{x} \in \Lambda$, $D_{\Lambda,\sigma}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda)}$. In the following lemma, we review several well-known facts about discrete Gaussian distribution:

**Lemma 2 ([19][43]).** *Let* $n$ *and* $q \geq 2$ *be integers. Let* $m \geq 2n \log q$*, and* $\sigma \geq \omega(\sqrt{\log m})$*.*

1. *For all but a* $2q^{-n}$ *fraction of all* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, for* $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$*, the distribution of* $\mathbf{u} = \mathbf{A} \cdot \mathbf{x} \bmod q$ *is statistically close to uniform over* $\mathbb{Z}_q^n$*. Moreover, the conditional distribution of* $\mathbf{x}$ *given* $\mathbf{u}$ *is* $D_{\Lambda^{\perp}_{\mathbf{u}}(\mathbf{A}),\sigma}$*.*
2. *For* $\beta = \lceil \sigma \cdot \log m \rceil$*, and* $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$*,* $\Pr\big[\, \|\mathbf{x}\|_\infty > \beta \,\big]$ *is negligible.*
3. *The min-entropy of* $D_{\mathbb{Z}^m,\sigma}$ *is at least* $m - 1$*.*

We now recall the results about two fundamental tools in lattice-based cryptography: the trapdoor generation and the preimage sampling algorithms. The algorithms stated in the following lemma are improvements of those in the literature [2,19,42,3].

**Lemma 3 ([36]).** *Given integers $n \geq 1$, $q \geq 2$, and $m \geq 2n \log q$. There is a* PPT *algorithm* $\mathsf{GenTrap}(n, m, q)$ *that outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a trapdoor* $\mathbf{R_A}$*, such that the distribution of* $\mathbf{A}$ *is* $\mathrm{negl}(n)$*-far from uniform. Moreover, for any vector* $\mathbf{u}$ *in the image of* $\mathbf{A}$ *and* $\sigma = \omega(\sqrt{n \log q \log n})$*, there is a* PPT *algorithm* $\mathsf{SampleD}(\mathbf{R_A}, \mathbf{A}, \mathbf{u}, \sigma)$ *that outputs* $\mathbf{x} \in \mathbb{Z}^m$ *sampled from the distribution* $D_{\mathbb{Z}^m,\sigma}$*, conditioned on the event that* $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \mod q$*.*

**The KTX String Commitment Scheme.** Kawachi et al. [22] constructed a string commitment scheme $\mathrm{COM} : \{0,1\}^* \times \{0,1\}^{\overline{m}/2} \rightarrow \mathbb{Z}_q^n$, such that:

- If $\overline{m} > 2n(1+\delta) \log q$ for some positive constant $\delta$, then COM is statistically hiding.
- If the $\mathsf{SIS}^{\infty}_{n,\overline{m},q,1}$ problem is hard, then COM is computationally binding.

In this paper, we will extensively employ the KTX commitment scheme. We implicitly choose $\overline{m}$ sufficiently large, e.g., $\overline{m} = 4n \log q$, to make COM statistically hiding.

# 3 Preparations

This section describes the parameters and techniques that will be used in our scheme.

## 3.1 Parameters

Our group signature scheme involves two main parameters: a security parameter $n$ and a maximum expected number of group users $N = 2^\ell \in \mathsf{poly}(n)$. Given $n$, we fix the other scheme parameters as in Table 1.

| **Parameter** | **Value** or **Asymptotic bound** |
|---|---|
| Prime modulus $q$ | $\omega(n^2 \log n)$ |
| Dimension $m$ | $\geq 2n \log q$ |
| Gaussian parameter $\sigma$ | $\omega(\sqrt{n \log q \log n})$ |
| Integer norm bound $\beta$ | $\lceil \sigma \cdot \log m \rceil$ s.t. $(4\beta + 1)^2 \leq q$ |
| Number of 'decompositions' $p$ | $\lfloor \log \beta \rfloor + 1$ |
| Sequence of integers<br>$\beta_1, \beta_2, \beta_3, \ldots, \beta_p$ | $\beta_1 = \lceil \beta/2 \rceil; \beta_2 = \lceil (\beta - \beta_1)/2 \rceil$<br>$\beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil; \ldots; \beta_p = 1$ |
| Number of protocol repetitions $t$ | $\omega(\log n)$ |

**Table 1.** Parameters of our VLR group signature scheme. The sequence $\beta_1, \beta_2, \ldots, \beta_p$ satisfies $\sum_{j=1}^{p} \beta_j = \beta$, and every integer in the interval $[-\beta, \beta]$ can be efficiently expressed as a subset sum of elements in the set $\{\pm\beta_1, \pm\beta_2, \ldots, \pm\beta_p\}$.

### 3.2 Some Specific Sets

We now define some specific sets of vectors and permutations that will be extensively used throughout this work. First, we denote by $\mathsf{B}_{3m}$ the set of all vectors in $\{-1,0,1\}^{3m}$ having exactly $m$ coordinates $-1$; $m$ coordinates 0; and $m$ coordinates 1. Given a binary string $d = d[1]\ldots d[\ell] \in \{0,1\}^\ell$, we define the following two sets of vectors:

- $\mathsf{Secret}_\beta(d)$: The set of all $\mathbf{x} = \big(\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ consisting of $2\ell+1$ *blocks* of size $m$, such that $\|\mathbf{x}\|_\infty \leq \beta$, and the following $\ell$ blocks are *zero-blocks* $\mathbf{0}^m$: $\mathbf{x}_1^{1-d[1]},\ldots,\mathbf{x}_\ell^{1-d[\ell]}$.
- $\mathsf{SecretExt}(d)$: The set of all $\mathbf{x} = \big(\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1\big) \in \{-1,0,1\}^{(2\ell+1)3m}$ consisting of $2\ell+1$ *blocks* of size $3m$, such that the $\ell+1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{d[1]},\ldots,\mathbf{x}_\ell^{d[\ell]}$ are elements of $\mathsf{B}_{3m}$, and the remaining $\ell$ blocks $\mathbf{x}_1^{1-d[1]},\ldots,\mathbf{x}_\ell^{1-d[\ell]}$ are *zero-blocks* $\mathbf{0}^{3m}$.

Given a vector $\mathbf{x} = \big(\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)3m}$ consisting of $2\ell+1$ blocks of size $3m$, we define two sets of permutations of $\mathbf{x}$:

- The set $\mathcal{S}$ of all permutations that keep the arrangement of the blocks. Specifically, if $\pi \in \mathcal{S}$, then
$$\pi(\mathbf{x}) = \big(\tau_0(\mathbf{x}_0)\|\tau_1^0(\mathbf{x}_1^0)\|\tau_1^1(\mathbf{x}_1^1)\|\ldots\|\tau_\ell^0(\mathbf{x}_\ell^0)\|\tau_\ell^1(\mathbf{x}_\ell^1)\big),$$
where $\tau_0,\tau_1^0,\tau_1^1,\ldots,\tau_\ell^0,\tau_\ell^1 \in \mathcal{S}_{3m}$. (Hereunder, $\mathcal{S}_{3m}$ denotes the symmetric group of all permutations of $3m$ elements.)
- The set $\mathcal{T} = \{T_e \big| \, e \in \{0,1\}^\ell\}$, where for $e = e[1]\ldots e[\ell]$, $T_e \in \mathcal{T}$ *rearranges* the blocks as follows:
$$T_e(\mathbf{x}) = \big(\mathbf{x}_0\|\mathbf{x}_1^{e[1]}\|\mathbf{x}_1^{1-e[1]}\|\ldots\|\mathbf{x}_\ell^{e[\ell]}\|\mathbf{x}_\ell^{1-e[\ell]}\big).$$

In particular, given $d, c \in \{0,1\}^\ell$, $\pi \in \mathcal{S}$, and $\mathbf{x} \in \mathbb{Z}^{(2\ell+1)3m}$, it can be checked that:

$$\mathbf{x} \in \mathsf{SecretExt}(d) \Leftrightarrow \pi(\mathbf{x}) \in \mathsf{SecretExt}(d) \Leftrightarrow T_c \circ \pi(\mathbf{x}) \in \mathsf{SecretExt}(d \oplus c), \quad (1)$$

where $\oplus$ denotes the (bit-wise) addition operation modulo 2.

### 3.3 The Decomposition - Extension Technique

Ling et al. [30] proposed a Stern-type zero-knowledge proof of knowledge for the $\mathsf{ISIS}^\infty_{n,m,q,\beta}$ problem that enjoys a strong security guarantee: the best way to break their protocol is to solve the underlying $\mathsf{ISIS}$ problem. They achieve this feature by using a versatile Decomposition-Extension framework. Adapting their technique, we construct the following procedures:

**Elementary Decomposition.** On input a vector $\mathbf{v} = (v_1, v_2, \ldots, v_m) \in \mathbb{Z}^m$ such that $\|\mathbf{v}\|_\infty \leq \beta$, the procedure $\mathsf{EleDec}$ outputs $p = \lfloor \log \beta \rfloor + 1$ vectors $\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_p \in \{-1,0,1\}^m$, such that $\sum_{j=1}^p \beta_j \cdot \widetilde{\mathbf{w}}_j = \mathbf{v}$. This procedure works as follows:

1. For each $i \in [m]$, express $v_i$ as $v_i = \beta_1 \cdot v_{i,1} + \beta_2 \cdot v_{i,2} + \ldots + \beta_p \cdot v_{i,p}$, where $\forall j \in [p] : \; v_{i,j} \in \{-1, 0, 1\}$. It was noted in [30] that for $\beta_1, \beta_2, \ldots, \beta_p$ given in Table 1, this step can easily be done.
2. For each $j \in [p]$, let $\widetilde{\mathbf{w}}_j := (v_{1,j}, v_{2,j}, \ldots, v_{m,j}) \in \{-1, 0, 1\}^m$. Output $\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_p$.

**Elementary Extension.** On input a vector $\widetilde{\mathbf{w}} \in \{-1, 0, 1\}^m$, the procedure $\mathsf{EleExt}$ extends $\widetilde{\mathbf{w}}$ to a vector $\mathbf{w} \in \mathsf{B}_{3m}$. This procedure works as follows:

1. Let $\lambda^{(-1)}$, $\lambda^{(0)}$ and $\lambda^{(1)}$ be the numbers of coordinates of $\widetilde{\mathbf{w}}$ that equal to $-1$, $0$, and $1$ respectively.
2. Pick a random vector $\widehat{\mathbf{w}} \in \{-1, 0, 1\}^{2m}$ that has exactly $(m - \lambda^{(-1)})$ coordinates $-1$, $(m - \lambda^{(0)})$ coordinates $0$, and $(m - \lambda^{(1)})$ coordinates $1$. Output vector $\mathbf{w} = \big(\widetilde{\mathbf{w}} \| \widehat{\mathbf{w}}\big) \in \mathsf{B}_{3m}$.

**Witness Decomposition and Extensions.** On input $\mathbf{x} \in \mathsf{Secret}_\beta(d)$ for some $d = d[1] \ldots d[\ell] \in \{0,1\}^\ell$, the procedure $\mathsf{WitnessDE}$ outputs $p$ vectors $\mathbf{z}_1, \ldots \mathbf{z}_p \in \mathsf{SecretExt}(d)$. This procedure works as follows:

1. Write vector $\mathbf{x}$ as the concatenation of $2\ell + 1$ blocks of size $m$, namely: $\mathbf{x} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big)$.
2. Run $\mathsf{EleDec}$ on each of the $\ell + 1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]}$ to obtained $(\ell + 1)p$ decomposed vectors. Then run $\mathsf{EleExt}$ on each of the decomposed vectors to obtain $(\ell + 1)p$ vectors in $\mathsf{B}_{3m}$, denoted respectively by
$$\{\mathbf{w}_{0,j}\}_{j=1}^p, \{\mathbf{w}_{1,j}^{d[1]}\}_{j=1}^p, \ldots, \{\mathbf{w}_{\ell,j}^{d[\ell]}\}_{j=1}^p.$$
3. Create $\ell p$ *zero-vectors* of dimension $3m$, and denote them by
$$\{\mathbf{w}_{1,j}^{1-d[1]}\}_{j=1}^p, \ldots, \{\mathbf{w}_{\ell,j}^{1-d[\ell]}\}_{j=1}^p.$$
4. For each $j \in [p]$, let $\mathbf{z}_j = \big(\mathbf{w}_{0,j} \| \mathbf{w}_{1,j}^0 \| \mathbf{w}_{1,j}^1 \| \ldots \| \mathbf{w}_{\ell,j}^0 \| \mathbf{w}_{\ell,j}^1\big)$.
Output $\mathbf{z}_1, \ldots, \mathbf{z}_p \in \mathsf{SecretExt}(d)$.

**Matrix Extension.** On input matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, the following procedure $\mathsf{MatrixExt}$ outputs matrix $\mathbf{A}^* \in \mathbb{Z}_q^{n \times (2\ell+1)3m}$:

1. Write $\mathbf{A}$ as the concatenation of $2\ell + 1$ component-matrices in $\mathbb{Z}_q^{n \times m}$.
2. Append $2m$ *zero-columns* to each of the component-matrices, then output the extended matrix $\mathbf{A}^*$.

In particular, let $\{\mathbf{z}_j\}_{j=1}^p \leftarrow \mathsf{WitnessDE}(\mathbf{x})$ and $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$ then we have $\mathbf{A} \cdot \mathbf{x} = \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j)$. We illustrate our Decomposition-Extension technique in Figure 1.

Therefore, in the protocol in Section 4, in order to prove that $\mathbf{x} \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, one can instead prove that:

$$\begin{cases} \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j) = \mathbf{u} \bmod q, \\ \forall j \in [p], \pi \in \mathcal{S}, e \in \{0,1\}^\ell : T_e \circ \pi(\mathbf{z}_j) \in \mathsf{SecretExt}(d \oplus e), \end{cases}$$

where the latter relation follows from the fact that $\mathbf{z}_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, and from (1).

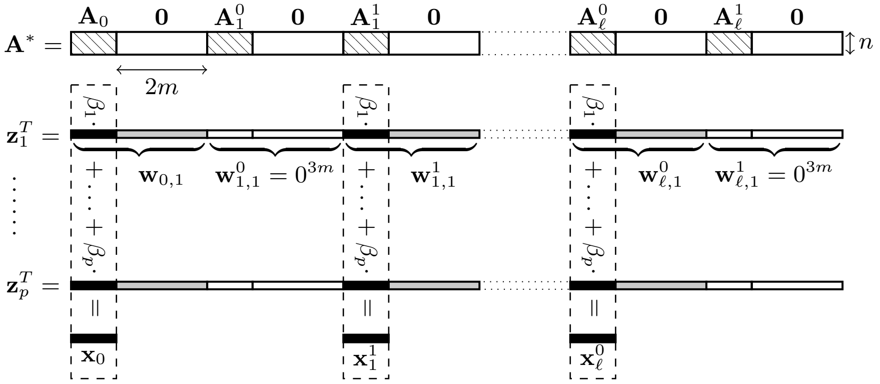

**Fig. 1.** An illustration of our Decomposition-Extension technique, where the first bit of $d$ is 1 and its last bit is 0. We denote by ▬▭ an element of $\mathsf{B}_{3m}$. After performing Decomposition-Extension, one has that $\mathbf{z}_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, and $\mathbf{A}^* \cdot \left(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j\right) = \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.

## 4 The Underlying Interactive Protocol

The main building block of our VLR group signature scheme is a Stern-like [48] interactive argument system allowing a signer to convince the verifier in zero-knowledge that:

1. The signer is a certified group member: he possesses a signature on his secret index, with respect to the Bonsai tree signature scheme [14].
2. The signer's revocation token is correctly committed via an $\mathsf{LWE}$ function.

In Section 5, the protocol is repeated $t = \omega(\log n)$ times to make the soundness error negligibly small, and then is transform to a signature scheme via Fiat-Shamir heuristic. The interactive protocol is summarized as follows:

- The public input consists of matrix $\mathbf{A} = \left[\mathbf{A}_0 \middle| \mathbf{A}_1^0 \middle| \mathbf{A}_1^1 \middle| \ldots \middle| \mathbf{A}_\ell^0 \middle| \mathbf{A}_\ell^1\right] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, vector $\mathbf{u} \in \mathbb{Z}_q^n$, matrix $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$ and vector $\mathbf{b} \in \mathbb{Z}_q^m$.
- The prover's witness consists of vector $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathsf{Secret}_\beta(d)$ for some secret $d \in \{0,1\}^\ell$; and vector $\mathbf{e} \in \mathbb{Z}^m$.
- The prover's goal is to convince the verifier in zero-knowledge that:
  1. $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$ and $\mathbf{x} \in \mathsf{Secret}_\beta(d)$.
  2. $\|\mathbf{e}\|_\infty \leq \beta$ and $\mathbf{B} \cdot (\mathbf{A}_0 \cdot \mathbf{x}_0) + \mathbf{e} = \mathbf{b} \bmod q$.

Prior to the interaction, both the prover and the verifier compute the following public matrices: $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$; $\mathbf{B}^* = \mathbf{B} \cdot \mathbf{A}_0 \in \mathbb{Z}_q^{m \times m}$; and $\mathbf{I}^* \in \{0,1\}^{m \times 3m}$, which is the matrix obtained by appending $2m$ zero-columns to the identity matrix of order $m$.

Meanwhile, the prover applies the Decomposition-Extension technique on his witness vectors:

- Let $\mathbf{z}_1, \ldots, \mathbf{z}_p \leftarrow \mathsf{WitnessDE}(\mathbf{x})$, and for each $j \in [p]$, let $\mathbf{z}_{j,0} = \mathsf{Parse}(\mathbf{z}_j, 1, m)$.
- Let $\tilde{\mathbf{e}}_1, \ldots, \tilde{\mathbf{e}}_p \leftarrow \mathsf{EleDec}(\mathbf{e})$, then for each $i \in [p]$, let $\mathbf{e}_i \leftarrow \mathsf{EleExt}(\tilde{\mathbf{e}}_i)$.

In the protocol, the prover instead convinces the verifier that he knows secret vectors $\mathbf{z}_1, \ldots, \mathbf{z}_p \in \mathsf{SecretExt}(d)$ and $\mathbf{e}_1, \ldots, \mathbf{e}_p \in \mathsf{B}_{3m}$ such that:

$$\begin{cases} \mathbf{A}^* \cdot \big(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j\big) = \mathbf{u} \bmod q; \\ \mathbf{B}^* \cdot \big(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_{j,0}\big) + \mathbf{I}^* \cdot \big(\sum_{j=1}^{p} \beta_j \cdot \mathbf{e}_j\big) = \mathbf{b} \bmod q. \end{cases}$$

### 4.1 Description of the Protocol

The protocol follows Stern's approach for three-pass zero-knowledge identification schemes [48,30]. Let $\mathsf{COM}$ be the statistically hiding and computationally binding string commitment scheme from [22]. The prover and the verifier interact as follows.

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

$$\begin{cases} c \xleftarrow{\$} \{0,1\}^{\ell}; \\ \pi_{z,1}, \ldots, \pi_{z,p} \xleftarrow{\$} \mathcal{S}; \;\; \pi_{e,1}, \ldots, \pi_{e,p} \xleftarrow{\$} \mathcal{S}_{3m}; \\ \mathbf{r}_{z,1}, \ldots, \mathbf{r}_{z,p} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}; \;\; \mathbf{r}_{e,1}, \ldots, \mathbf{r}_{e,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}. \end{cases}$$

   For each $j \in [p]$, let $\mathbf{r}_{j,0} = \mathsf{Parse}(\mathbf{r}_{z,j}, 1, m)$. Then it and sends the commitment $\mathrm{CMT} = \big(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\big)$ to the verifier, where

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}\big(c, \; \{\pi_{z,j}, \pi_{e,j}\}_{j=1}^{p}, \; \mathbf{A}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{z,j}); \\ \qquad\qquad \mathbf{B}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{j,0}) + \mathbf{I}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_{e,j}); \; \rho_1\big), \\ \mathbf{c}_2 = \mathsf{COM}\big(\{\mathrm{T}_c \circ \pi_{z,j}(\mathbf{r}_{z,j}), \; \pi_{e,j}(\mathbf{r}_{e,j})\}_{j=1}^{p}; \; \rho_2\big), \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathrm{T}_c \circ \pi_{z,j}(\mathbf{z}_j + \mathbf{r}_{z,j}), \; \pi_{e,j}(\mathbf{e}_j + \mathbf{r}_{e,j})\}_{j=1}^{p}; \; \rho_3\big). \end{cases} \tag{2}$$

2. **Challenge:** The verifier sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$ to the prover.
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 $\mathbf{v}_{z,j} = \mathrm{T}_c \circ \pi_{z,j}(\mathbf{z}_j)$; $\mathbf{w}_{z,j} = \mathrm{T}_c \circ \pi_{z,j}(\mathbf{r}_{z,j})$; $\mathbf{v}_{e,j} = \pi_{e,j}(\mathbf{e}_j)$; $\mathbf{w}_{e,j} = \pi_{e,j}(\mathbf{r}_{e,j})$. Send

$$\mathrm{RSP} = \big(d_1, \; \{\mathbf{v}_{z,j}, \mathbf{w}_{z,j}, \mathbf{v}_{e,j}, \mathbf{w}_{e,j}\}_{j=1}^{p}, \; \rho_2, \rho_3\big). \tag{3}$$

   - Case $Ch = 2$: Let $d_2 = c$. For each $j \in [p]$, let $\phi_{z,j} = \pi_{z,j}$; $\phi_{e,j} = \pi_{e,j}$; $\mathbf{s}_{z,j} = \mathbf{z}_j + \mathbf{r}_{z,j}$; $\mathbf{s}_{e,j} = \mathbf{e}_j + \mathbf{r}_{e,j}$. Send

$$\mathrm{RSP} = \big(d_2, \; \{\phi_{z,j}, \phi_{e,j}, \mathbf{s}_{z,j}, \mathbf{s}_{e,j}\}_{j=1}^{p}, \; \rho_1, \rho_3\big). \tag{4}$$

- Case $Ch = 3$: Let $d_3 = c$. For each $j \in [p]$, let $\psi_{z,j} = \pi_{z,j}$; $\psi_{e,j} = \pi_{e,j}$; $\mathbf{h}_{z,j} = \mathbf{r}_{z,j}$; $\mathbf{h}_{e,j} = \mathbf{r}_{e,j}$. Send
$$\mathrm{RSP} = \big(d_3,\ \{\psi_{z,j}, \psi_{e,j}, \mathbf{h}_{z,j}, \mathbf{h}_{e,j}\}_{j=1}^{p},\ \rho_1, \rho_2\big). \tag{5}$$

**Verification:** Receiving the response RSP, the verifier proceeds as follows:

- Case $Ch = 1$: Parse RSP as in (3). Check that $\forall j \in [p]:\ \mathbf{v}_{z,j} \in \mathsf{SecretExt}(d_1)$ and $\mathbf{v}_{e,j} \in \mathsf{B}_{3m}$ and that:
$$\begin{cases} \mathbf{c}_2 = \mathsf{COM}\big(\{\mathbf{w}_{z,j},\ \mathbf{w}_{e,j}\}_{j=1}^{p};\ \rho_2\big) \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathbf{v}_{z,j} + \mathbf{w}_{z,j},\ \mathbf{v}_{e,j} + \mathbf{w}_{e,j}\}_{j=1}^{p};\ \rho_3\big). \end{cases}$$
- Case $Ch = 2$: Parse RSP as in (4). For each $j \in [p]$, let $\mathbf{s}_{j,0} = \mathsf{Parse}(\mathbf{s}_{z,j}, 1, m)$. Check that:
$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}\big(d_2,\ \{\phi_{z,j}, \phi_{e,j}\}_{j=1}^{p},\ \mathbf{A}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{z,j}) - \mathbf{u}; \\ \qquad\qquad \mathbf{B}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{j,0}) + \mathbf{I}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_{e,j}) - \mathbf{b};\ \rho_1\big), \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}),\ \phi_{e,j}(\mathbf{s}_{e,j})\}_{j=1}^{p};\ \rho_3\big). \end{cases}$$
- Case $Ch = 3$: Parse RSP as in (5). For each $j \in [p]$, let $\mathbf{h}_{j,0} = \mathsf{Parse}(\mathbf{h}_{z,j}, 1, m)$. Check that:
$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}\big(d_3,\ \{\psi_{z,j}, \psi_{e,j}\}_{j=1}^{p},\ \mathbf{A}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{z,j}); \\ \qquad\qquad \mathbf{B}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{j,0}) + \mathbf{I}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{h}_{e,j});\ \rho_1\big); \\ \mathbf{c}_2 = \mathsf{COM}\big(\{\mathrm{T}_{d_3} \circ \psi_{z,j}(\mathbf{h}_{z,j}),\ \psi_{e,j}(\mathbf{h}_{e,j})\}_{j=1}^{p};\ \rho_2\big). \end{cases}$$

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

### 4.2 Analysis of the Protocol

The following theorem summarizes the properties of the given interactive protocol.

**Theorem 1.** *Let* COM *be a statistically hiding and computationally binding string commitment scheme. Then the interactive protocol described in Section 4.1 is a zero-knowledge argument of knowledge with perfect completeness, soundness error* $2/3$*, and communication cost* $\ell \cdot \widetilde{\mathcal{O}}(n)$*. In particular:*

- *There exists an efficient simulator that, on public input* $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{b})$*, 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* $(\mathrm{RSP}^{(1)}, \mathrm{RSP}^{(2)}, \mathrm{RSP}^{(3)})$ *corresponding to all* 3 *possible values of the challenge* $Ch$*, outputs a pair of vector* $(\mathbf{y}, \mathbf{e}') \in \mathbb{Z}^{(2\ell+1)m} \times \mathbb{Z}^m$ *such that:*

1. $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big) \in \mathsf{Secret}_\beta(d)$ *for some* $d \in \{0,1\}^\ell$, *and* $\mathbf{A} \cdot \mathbf{y} = \mathbf{u} \bmod q$.
2. $\|\mathbf{e}'\|_\infty \leq \beta$ *and* $\mathbf{B} \cdot (\mathbf{A}_0 \cdot \mathbf{y}_0) + \mathbf{e}' = \mathbf{b} \bmod q$.

The proof of Theorem 1 employs standard simulation and extraction techniques for Stern-like protocol [48,22,30], and is given in Appendix A.

## 5 The VLR Group Signature Scheme

In this section we first describe our lattice-based VLR group signature scheme, and then we prove that the scheme satisfies the requirements defined in Section 2.1: correctness, selfless-anonymity and traceability. The parameters of the scheme are as specified in Table 1.

### 5.1 Description of the Scheme

**Keys Generation.** The randomized algorithm $\mathsf{KeyGen}(n, N)$, works as follows:

1. Run $\mathsf{GenTrap}(n, m, q)$ to get $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ and trapdoor $\mathbf{R}$.
2. Sample $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ for all $b \in \{0,1\}$ and $i \in [\ell]$. Then define the matrix
$$\mathbf{A} = \big[\mathbf{A}_0 \big| \mathbf{A}_1^0 \big| \mathbf{A}_1^1 \big| \ldots \big| \mathbf{A}_\ell^0 \big| \mathbf{A}_\ell^1\big] \in \mathbb{Z}_q^{n \times (2\ell+1)m}.$$
3. For group user with index $d \in \{0, 1, \ldots, N-1\}$, let $d[1] \ldots d[\ell] \in \{0,1\}^\ell$ denote the binary representation of $d$, and do the following:
   (a) Sample $\mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and let $\mathbf{z} = \sum_{i=1}^{\ell} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]} \bmod q$. Then sample $\mathbf{x}_0 \in \mathbb{Z}^m$ with $\mathbf{x}_0 \hookleftarrow \mathsf{SampleD}\big(\mathbf{R}, \mathbf{A}_0, \mathbf{u} - \mathbf{z}, \sigma\big)$.
   Next, let $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$ be zero-vectors $\mathbf{0}^m$, and define
   $$\mathbf{x}^{(d)} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}.$$
   If $\|\mathbf{x}^{(d)}\|_\infty \leq \beta$ then go to step (3b); else, repeat step (3a).
   (b) Let the user secret key be $\mathsf{gsk}[d] = \mathbf{x}^{(d)}$, and the revocation token be $\mathsf{grt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \in \mathbb{Z}_q^n$.
4. Finally, the algorithm outputs $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$, where
$$\mathsf{gpk} = (\mathbf{A}, \mathbf{u}); \;\; \mathsf{gsk} = \big(\mathsf{gsk}[0], \mathsf{gsk}[1], \ldots, \mathsf{gsk}[N-1]\big);$$
$$\mathsf{grt} = \big(\mathsf{grt}[0], \mathsf{grt}[1], \ldots, \mathsf{grt}[N-1]\big).$$

*Remark 2.* We have some observations on the behaviour of the above key generation algorithm:

- By Lemma 3, the distribution of matrix $\mathbf{A}_0$ generated by $\mathsf{GenTrap}(n, m, q)$ is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$. Thus, the distribution of $\mathsf{gpk}$ output by $\mathsf{KeyGen}(n, N)$ is statistically close to uniform over $\mathbb{Z}_q^{n \times (2\ell+1)m} \times \mathbb{Z}_q^n$. We note that the pair $(\mathbf{A}, \mathbf{u})$ resembles the Bonsai tree structure [14], where $\mathbf{A}_0$ is the "root" of the tree.

- In Step (3a), each coordinate of vector $\mathbf{x}^{(d)}$ is either 0 or distributed according to the distribution $D_{\mathbb{Z},\sigma}$ (see Lemma 3 regarding the output distribution of algorithm $\mathsf{SampleD}$). By setting $\beta = \lceil \sigma \cdot \log m \rceil$, we ensure that $\|\mathbf{x}^{(d)}\|_\infty \leq \beta$ with overwhelming probability (see Lemma 2). Thus, the event that Step (3a) needs to be repeated only occurs with negligible probability.
- The secret key $\mathbf{x}^{(d)}$ of group user with index $d$ satisfies $\mathbf{A} \cdot \mathbf{x}^{(d)} = \mathbf{u} \bmod q$, and $\mathbf{x}^{(d)} \in \mathsf{Secret}_\beta(d)$.
- By Lemma 2, the distribution of each user revocation token $\mathsf{grt}[d]$ is statistically close to uniform over $\mathbb{Z}_q^n$. The trivial requirement is that the revocation tokens of two different group users must be different. In the very rare event of conflict (i.e., there exist $d_1, d_2 \in \{0, \ldots, N-1\}$ such that $d_2 > d_1$ and $\mathsf{grt}[d_1] = \mathsf{grt}[d_2]$), the algorithm simply re-samples the key and token for user with index $d_2$.

**Signing Algorithm.** Let $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$ and $\mathcal{G} : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ be random oracles. Let $\chi$ be a $\beta$-bounded distribution, e.g., $\chi = D_{\mathbb{Z}^m,\sigma}$. Given $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, to sign a message $M \in \{0,1\}^*$ using the secret key $\mathsf{gsk}[d] = \mathbf{x} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathsf{Secret}_\beta(d)$, the user runs the randomized algorithm $\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$, which performs the following steps:

1. Sample $\rho \xleftarrow{\$} \{0,1\}^n$ and let $\mathbf{B} = \mathcal{G}(\mathbf{A}, \mathbf{u}, M, \rho) \in \mathbb{Z}_q^{m \times n}$.
2. Sample $\mathbf{e} \hookleftarrow \chi^m$. (Note that $\|\mathbf{e}\|_\infty \leq \beta$ with overwhelming probability.) Then compute

$$\mathbf{b} = \mathbf{B} \cdot \mathsf{grt}[d] + \mathbf{e} \bmod q = \mathbf{B} \cdot (\mathbf{A}_0 \cdot \mathbf{x}_0) + \mathbf{e} \bmod q.$$

3. Generate a non-interactive zero-knowledge argument of knowledge $\Pi$ to prove the possession of a valid secret signing key, and that vector $\mathbf{b} \in \mathbb{Z}_q^m$ is honestly computed as above. This is done by repeating $t = \omega(\log n)$ times the basic protocol from Section 4 with public parameter $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{b})$ and prover's witness $(\mathbf{x}, \mathbf{e})$, and then making it non-interactive with the Fiat-Shamir heuristic as a triple $\Pi = \big(\{\mathrm{CMT}^{(k)}\}_{k=1}^t,\ \mathrm{CH},\ \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$, where

$$\mathrm{CH} = \big(\{Ch^{(k)}\}_{k=1}^t\big) = \mathcal{H}\big(M, \mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{b},\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big) \in \{1,2,3\}^t.$$

4. Output the group signature:

$$\Sigma = \big(M, \rho, \mathbf{b}, \Pi\big). \tag{6}$$

**Verification Algorithm.** On input group public key $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, a set of tokens $RL = \big\{\{\mathbf{u}_i\}_i\big\} \subset \mathbb{Z}_q^n$ whose cardinality is at most $N-1$, a message $M \in \{0,1\}^*$, and a purported group signature $\Sigma$ on $M$, the verifier runs the deterministic algorithm $\mathsf{Verify}(\mathsf{gpk}, RL, \Sigma, M)$, which performs the following steps:

1. Parse the signature $\Sigma$ as in (6), and compute $\mathbf{B} = \mathcal{G}(\mathbf{A}, \mathbf{u}, M, \rho) \in \mathbb{Z}_q^{m \times n}$.
2. If $\big(Ch^{(1)}, \ldots, Ch^{(t)}\big) = \mathcal{H}\big(M, \mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{b},\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big)$, then return $\mathsf{Invalid}$.

3. For $k = 1$ to $t$, run the verification of the protocol from Section 4 to check the validity of $\mathrm{RSP}^{(k)}$ with respect to $\mathrm{CMT}^{(k)}$ and $Ch^{(k)}$. If any of the verification conditions does not hold, then return $\mathsf{Invalid}$.
4. For each $\mathbf{u}_i \in RL$ compute vector $\mathbf{e}'_i = \mathbf{b} - \mathbf{B} \cdot \mathbf{u}_i \bmod q$. If there exists an index $i$ such that $\|\mathbf{e}'_i\|_\infty \leq \beta$, then return $\mathsf{Invalid}$.
5. Return $\mathsf{Valid}$.

### 5.2 Analysis of the Scheme

**Efficiency and Correctness.** The parameters in Table 1 are set so that all of the algorithms in the VLR group signature in Section 5.1 can be implemented in polynomial time. The group public key has bit-size $\ell \cdot \widetilde{\mathcal{O}}(n^2) = \log N \cdot \widetilde{\mathcal{O}}(n^2)$, while the group signatures have bit-size $\ell \cdot \widetilde{\mathcal{O}}(n) = \log N \cdot \widetilde{\mathcal{O}}(n)$. The revocation check, which corresponds to Step 4 of the signature verification algorithm, runs in linear time in the number of revoked users, as it seems unavoidable for secure VLR group signature schemes.

To demonstrate the correctness of the scheme, we will make use of the following lemma.

**Lemma 4.** *Let $\beta = \mathsf{poly}(n)$, $q \geq (4\beta + 1)^2$ and $m \geq 3n$. Then, over the randomness of $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, we have*

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

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

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

□

In particular, for parameters $q, m, \beta$ specified in Table 1, for uniformly random matrix $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$, and for non-zero vector $\mathbf{s} \in \mathbb{Z}_q^n$, we have $\|\mathbf{B} \cdot \mathbf{s}\|_\infty > 2\beta$ with overwhelming probability.

**Theorem 2.** *The VLR group signature scheme described in Section 5.1 is correct with overwhelming probability.*

*Proof.* According to the correctness requirement, we have to prove that for all $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, $\mathsf{gsk} = (\{\mathsf{gsk}[d]\}_{d=0}^{N-1})$, $\mathsf{grt} = (\{\mathsf{grt}[d]\}_{d=0}^{N-1})$ outputted by algorithm $\mathsf{KeyGen}(n, N)$, all $d \in \{0, 1, \ldots, N-1\}$, and all $M \in \{0,1\}^*$, we have:

$$\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid} \Leftrightarrow \mathsf{grt}[d] \notin RL.$$

We first remark that, with overwhelming probability, the honest signer can obtain a valid witness $(\mathbf{x}, \mathbf{e}) \in \mathsf{Secret}_\beta(d) \times [-\beta, \beta]^m$ to be used in the underlying argument system. Then, thanks to the perfect completeness of the latter protocol, the signature verification algorithm should not return $\mathsf{Invalid}$ after its first 3 steps. We further note that, in Step 4, the vector $\mathbf{e}_i'$, for every $i$, can be expressed as:

$$\mathbf{e}_i' = \mathbf{b} - \mathbf{B} \cdot \mathbf{u}_i = \mathbf{B} \cdot \mathsf{grt}[d] + \mathbf{e} - \mathbf{B} \cdot \mathbf{u}_i = \mathbf{B} \cdot (\mathsf{grt}[d] - \mathbf{u}_i) + \mathbf{e} \bmod q.$$

Now, suppose that the verification algorithm outputs $\mathsf{Valid}$, i.e., we have that $\|\mathbf{e}_i'\|_\infty \leq \beta$, for all $i$. We will show that $\mathsf{grt}[d] \notin RL$. Indeed, if there exists index $i$ such that $\mathsf{grt}[d] = \mathbf{u}_i$, then $\mathbf{e}_i' = \mathbf{e}$. Since $\|\mathbf{e}_i'\|_\infty = \|\mathbf{e}\|_\infty \leq \beta$, the signature should not pass Step 4 of the verification procedure, which yields a contradiction.

Next, suppose that $\mathsf{grt}[d] \notin RL$. Then, for every $i$, vector $\mathbf{s}_i := \mathsf{grt}[d] - \mathbf{u}_i \bmod q$ is non-zero. We will demonstrate that, in this case, the verification algorithm outputs $\mathsf{Valid}$ with overwhelming probability. It suffices to show that $\mathbf{e}_i'$ has infinity norm larger than $\beta$, for all $i$. Indeed, on one hand, Lemma 4 implies that $\|\mathbf{B} \cdot \mathbf{s}_i\|_\infty > 2\beta$ with overwhelming probability. On the other hand, $\|\mathbf{B} \cdot \mathbf{s}_i\|_\infty \leq \|\mathbf{e}_i'\|_\infty + \|\mathbf{e}\|_\infty \leq \|\mathbf{e}_i'\|_\infty + \beta$. Hence, except for a negligible probability, we have $\|\mathbf{e}_i'\|_\infty > 2\beta - \beta = \beta$. □

**Selfless-Anonymity.** We now prove that our VLR group signature scheme is selfless-anonymous.

**Theorem 3.** *In the random oracle model, the VLR group signature scheme in Section 5.1 is selfless-anonymous under the* $\mathsf{LWE}_{n,q,\chi}$ *assumption.*

Via the reduction from $\mathsf{SIVP}$ to $\mathsf{LWE}$, the selfless anonymity of the scheme can be based on the worst-case hardness of $\mathsf{SIVP}_\gamma$, where $\gamma = \widetilde{\mathcal{O}}(n^{2.5})$, for the given setting of parameters.

*Proof.* We define a sequence of hybrid games $G_0^{(0)}$, $G_1^{(0)}$, $G_2^{(0)}$, $G_3$, $G_2^{(1)}$, $G_1^{(1)}$, $G_0^{(1)}$, such that game $G_0^{(c)}$, for $c \in \{0, 1\}$, is the original selfless-anonymity game (see Section 2) where the bit chosen by the challenger is $c$. We then prove that these games are indistinguishable, based on the zero-knowledge property of the underlying argument system and the hardness of the $\mathsf{LWE}_{n,q,\chi}$ problem. The selfless-anonymity of our scheme then follows from the fact that game $G_3$ is independent of the bit $c$.

**Game $G_0^{(c)}$:**

1. Run $\mathsf{KeyGen}(n, N)$ to obtain

$$\mathsf{gpk} = (\mathbf{A}, \mathbf{u});\ \ \mathsf{gsk} = \big(\mathsf{gsk}[0], \mathsf{gsk}[1], \ldots, \mathsf{gsk}[N-1]\big);$$
$$\mathsf{grt} = \big(\mathsf{grt}[0], \mathsf{grt}[1], \ldots, \mathsf{grt}[N-1]\big).$$

Set $RL := \emptyset$, $\mathsf{Corrupted} := \emptyset$, and give $\mathsf{gpk}$ to the adversary $\mathcal{A}$.

2. If $\mathcal{A}$ queries the signature on any message $M$ by user of index $d$, return $\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d], M\big)$. If $\mathcal{A}$ queries the corruption of user of index $d$, set $\mathsf{Corrupted} := \mathsf{Corrupted} \cup \{d\}$, and return $\mathsf{gsk}[d]$. If $\mathcal{A}$ queries the revocation of user $d$, set $RL := RL \cup \{\mathsf{grt}[d]\}$, and return $\mathsf{grt}[d]$.
3. $\mathcal{A}$ outputs a message $M^*$ and $d_0, d_1$ such that

$$d_0, d_1 \notin \mathsf{Corrupted} \text{ and } \mathsf{grt}[d_0], \mathsf{grt}[d_1] \notin RL.$$

4. Generate a legitimate signature

$$\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d_c], M^*\big) = \big(M^*, \rho, \mathbf{b}, \Pi\big),$$

and return $\Sigma$ to $\mathcal{A}$.
5. $\mathcal{A}$ can still make queries as before, but it is not allowed to ask for $\mathsf{gsk}[d_0]$, $\mathsf{gsk}[d_1]$, $\mathsf{grt}[d_0]$ and $\mathsf{grt}[d_1]$.
6. Finally $\mathcal{A}$ outputs a bit $c'$.

**Game $G_1^{(c)}$:**
In this game, we make the following modification with respect to Game $G_0^{(c)}$: In Step 4, instead of generating a legitimate signature, we simulate it as follows:

1. Run the first 2 steps of the signing algorithm in an honest manner, and obtain $\rho \in \{0,1\}^n$; $\mathbf{B} = \mathcal{G}(\mathbf{A}, \mathbf{u}, M^*, \rho)$; $\mathbf{e}_c \hookleftarrow \chi^m$; and $\mathbf{b} = \mathbf{B} \cdot \mathsf{grt}[d_c] + \mathbf{e}_c \in \mathbb{Z}_q^m$.
2. Simulate the non-interactive zero-knowledge argument $\Pi^*$. This is done by invoking the simulator of the argument system from Section 4 (see Theorem 1) for every repetition, and then programming the random oracle $\mathcal{H}$ accordingly. Since the underlying argument system is statistically zero-knowledge, the distribution of $\Pi^*$ is statistically close to that of the legitimate $\Pi$.
3. Output $\Sigma^* = (M^*, \rho, \mathbf{b}, \Pi^*)$.

It can be seen that $\Sigma^*$ is statistically close to the signature $\Sigma$ outputted by Step 4 of **Game** $G_0^{(c)}$. As a result, Game $G_1^{(c)}$ and Game $G_0^{(c)}$ are statistically indistinguishable.

**Game $G_2^{(c)}$:**
In this game, we introduce the following modification with respect to Game $G_1^{(c)}$. Recall that in $\mathbf{G}_1^{(c)}$, we have $\mathbf{b} = \mathbf{B} \cdot \mathsf{grt}[d_c] + \mathbf{e}_c \bmod q$, where the token $\mathsf{grt}[d_c]$, unknown to $\mathcal{A}$, is statistically close to uniform over $\mathbb{Z}_q^n$; $\mathbf{B}$ is uniformly random over $\mathbb{Z}_q^{m \times n}$; and $\mathbf{e}_b$ has entries sampled from the error distribution $\chi$. Now, we instead compute $\mathbf{b}$ as

$$\mathbf{b} = \mathbf{B} \cdot \mathbf{s}_c + \mathbf{e}_c \bmod q,$$

where $\mathbf{s}_c \xleftarrow{\$} \mathbb{Z}_q^n$. The rest of the game remains the same as in Game $G_1^{(c)}$. This way, the two games are statistically indistinguishable.

**Game $G_3$:**

Note that in the previous game, the pair $(\mathbf{B}, \mathbf{b})$ is a proper $\mathsf{LWE}_{n,q,\chi}$ instance (with $m$ samples). Thus, under the $\mathsf{LWE}_{n,q,\chi}$ assumption, its distribution is computationally close to the uniform distribution over $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$.

In this game, we make $\mathbf{b}$ truly uniform by sampling $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$ and setting $\mathbf{b} = \mathbf{v}$. The rest of the game is the same as in Game $G_2^{(c)}$. Clearly, the advantage of the adversary in this game is 0. Furthermore, if the $\mathsf{LWE}_{n,q,\chi}$ problem is hard, then $G_2^{(c)}$ and $G_3$ are computationally indistinguishable.

Finally, observe that we have established a chain of indistinguishable games

$$G_0^{(0)} \overset{\text{s}}{\approx} G_1^{(0)} \overset{\text{s}}{\approx} G_2^{(0)} \overset{\text{c}}{\approx} G_3 \overset{\text{c}}{\approx} G_2^{(1)} \overset{\text{s}}{\approx} G_1^{(1)} \overset{\text{s}}{\approx} G_0^{(1)},$$

which implies that the adversary's advantage in game $G_0^{(c)}$, for $c \in \{0, 1\}$, is negligible. This concludes the proof. □

**Traceability.** We now prove that, in the random oracle model, our VLR group signature scheme is traceable if the $\mathsf{SIS}^{\infty}_{n,(\ell+1)\cdot m,q,2\beta}$ problem is hard.

**Theorem 4.** *In the random oracle model, if there is a traceability adversary $\mathcal{A}$ with success probability $\epsilon$ and running time $T$, then there is an algorithm $\mathcal{F}$ that solves the $\mathsf{SIS}^{\infty}_{n,(\ell+1)\cdot m,q,2\beta}$ problem with success probability $\epsilon' > \big(1 - (7/9)^t\big) \cdot \frac{1}{2N}$, and running time $T' = 32 \cdot T \cdot q_{\mathcal{H}}/(\epsilon - 3^{-t}) + \mathsf{poly}(n, N)$, where $q_{\mathcal{H}}$ is the number of queries to the random oracle $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$.*

The results of Lemma 1 and Theorem 4 imply that our scheme is traceable in the random oracle model, based on the worst-case hardness of $\mathsf{SIVP}_\gamma$, where $\gamma = 2\beta \cdot \widetilde{\mathcal{O}}(\sqrt{n(\ell+1)m}) = \widetilde{\mathcal{O}}(n^{1.5})$.

*Proof.* First, suppose that adversary $\mathcal{A}$ can break with non-negligible probability the computational binding property of the commitment scheme $\mathsf{COM}$ employed by the underlying argument system. As mentioned earlier (see Section 2.2), we can use $\mathcal{A}$ to solve the $\mathsf{SIS}^{\infty}_{n,(\ell+1)\cdot m,q,2\beta}$ problem. Therefore, without loss of generality, we assume that $\mathsf{COM}$ is computationally binding.

We construct a PPT algorithm $\mathcal{F}$ solving the $\mathsf{SIS}^{\infty}_{n,(\ell+1)\cdot m,q,2\beta}$ problem with non-negligible probability, which works as follows.

$\mathsf{Challenge}$: Algorithm $\mathcal{F}$ is given as the $\mathsf{SIS}$ challenge a uniformly random matrix $\mathbf{C} = \big[\mathbf{C}_0\big|\mathbf{C}_1\big|\ldots\big|\mathbf{C}_\ell\big] \in \mathbb{Z}_q^{n\times(\ell+1)\cdot m}$. It wins the challenge if it can produce a non-zero vector $\mathbf{x} \in \mathbb{Z}^{(\ell+1)\cdot m}$ such that $\|\mathbf{x}\|_\infty \leq 2\beta$ and $\mathbf{C} \cdot \mathbf{x} = \mathbf{0} \mod q$.

$\mathsf{Setup}$: $\mathcal{F}$ performs the following steps:

1. Sample vector $\mathbf{z} = \big(\mathbf{z}_0\|\mathbf{z}_1\|\ldots\|\mathbf{z}_\ell\big) \in \mathbb{Z}^{(\ell+1)\cdot m}$, where each coordinate of $\mathbf{z}$ is sampled from $D_{\mathbb{Z},\sigma}$. If $\|\mathbf{z}\|_\infty > \beta$, then repeat the sampling. Otherwise, compute $\mathbf{u} = \mathbf{C} \cdot \mathbf{z} \bmod q$.
2. Run $\mathsf{TrapGen}(n, m, q)$ algorithm $\ell$ times, and let the outputs be

$$\big((\mathbf{F}_1, \mathbf{R}_1),\ (\mathbf{F}_2, \mathbf{R}_2), \ldots, (\mathbf{F}_\ell, \mathbf{R}_\ell)\big).$$

3. Pick a target index $d^* = d^*[1] \ldots d^*[\ell] \xleftarrow{\$} \{0,1\}^\ell$, and define matrix

$$\mathbf{A} = \big[\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1\big] \in \mathbb{Z}_q^{n\times(2\ell+1)\cdot m},$$

where $\mathbf{A}_0 = \mathbf{C}_0$, and for each $i \in [\ell]$: $\mathbf{A}_i^{d^*[i]} = \mathbf{C}_i$ and $\mathbf{A}_i^{1-d^*[i]} = \mathbf{F}_i$.
4. Define the secret key and revocation token for user $d^*$ as follows:
   - $\mathsf{gsk}[d^*] = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)\cdot m}$, where $\mathbf{x}_0 = \mathbf{z}_0$, $\forall i \in [\ell]$: $\mathbf{x}_i^{d^*[i]} = \mathbf{z}_i$ and $\mathbf{x}_i^{1-d^*[i]} = \mathbf{0}^m$,
   - $\mathsf{grt}[d^*] = \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q \in \mathbb{Z}_q^n$.
5. Generate the secret key and the revocation token for each user $d \neq d^*$, where $d = d[1]\ldots d[\ell]$, as follows:
   - Let $d[b]$ $(1 \leq b \leq \ell)$ be the first bit from the left where $d[b] \neq d^*[b]$. Since $d \neq d^*$, such $b$ must exist. It follows that $\mathbf{A}_b^{d[b]} = \mathbf{A}_b^{1-d^*[b]} = \mathbf{F}_b$.
   - Sample $\ell$ vectors $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_{b-1}^{d[b-1]}, \mathbf{x}_{b+1}^{d[b+1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, and let

   $$\mathbf{t}^{(d)} = \mathbf{u} - \big(\mathbf{A}_0 \cdot \mathbf{x}_0 + \sum_{i\in[\ell], i\neq b} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]}\big) \bmod q.$$

   - Sample $\mathbf{x}_b^{d[b]} \hookleftarrow \mathsf{SampleD}(\mathbf{R}_b, \mathbf{F}_b, \mathbf{t}^{(d)}, \sigma)$.
   - For each $i \in [\ell]$, let $\mathbf{x}_i^{1-d[i]} = \mathbf{0}^m$. Then form vector

   $$\mathbf{x}^{(d)} = \big(\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)\cdot m}.$$

   If the very rare event that $\|\mathbf{x}^{(d)}\|_\infty > \beta$ happens, then repeat the sampling. Otherwise, set $\mathsf{gsk}[d] = \mathbf{x}^{(d)}$ and $\mathsf{grt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q \in \mathbb{Z}_q^n$.
6. Let

$$\mathsf{gpk} = (\mathbf{A}, \mathbf{u});\ \ \mathsf{gsk} = \big(\mathsf{gsk}[0], \mathsf{gsk}[1], \ldots, \mathsf{gsk}[N-1]\big);$$
$$\mathsf{grt} = \big(\mathsf{grt}[0], \mathsf{grt}[1], \ldots, \mathsf{grt}[N-1]\big).$$

We note that, by construction, the distribution of $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$ is statistically close to that of the real scheme, and the choice of $d^*$ is hidden from the adversary. Algorithm $\mathcal{F}$ then gives $(\mathsf{gpk}, \mathsf{grt})$ to $\mathcal{A}$.

Queries: Algorithm $\mathcal{F}$ answers the queries of $\mathcal{A}$ as follows:

- **Corruption queries**: The corruption set $U$ is initially set to be empty. If $\mathcal{A}$ queries the secret key of any user $d \in \{0, \ldots, N-1\}$, then $\mathcal{F}$ adds $d$ to the corruption set $U$, and returns $\mathsf{gsk}[d]$.
- **Signatures queries**: If $\mathcal{A}$ queries signature of user $d$ on arbitrary message $M$, then $\mathcal{F}$ returns $\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d], M\big)$. Queries to the random oracles $\mathcal{H}$ and $\mathcal{G}$ are handled by consistently returning uniformly random values in their respective ranges. For each $\kappa \leq q_{\mathcal{H}}$, we let $r_\kappa$ denote the answer to the $\kappa$-th query.

Forgery: Eventually, $\mathcal{A}$ outputs a message $M^*$, a set of tokens $RL^*$ and a non-trivial forged signature

$$\Sigma^* = \big(M^*, \rho^*, \mathbf{b}^*, \{\mathrm{CMT}_i\}_{i=1}^t, \{Ch_i\}_{i=1}^t, \{\mathrm{RSP}_i\}_{i=1}^t\big),$$

such that $\mathsf{Verify}(\mathsf{gpk}, RL^*, \Sigma^*, M^*) = \mathsf{Valid}$, and the implicit tracing algorithm fails or traces to a user outside of the coalition $U \setminus RL^*$.

Let $\mathbf{B}^* = \mathcal{G}(\mathbf{A}, \mathbf{u}, M^*, \rho^*) \in \mathbb{Z}_q^{m\times n}$. Algorithm $\mathcal{F}$ then exploits the forgery as follows. First, one can argue that $\mathcal{A}$ must have queried $\mathcal{H}$ on input

$$\big(M^*, \mathbf{A}, \mathbf{u}, \mathbf{B}^*, \mathbf{b}^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big),$$

as otherwise, the probability that

$$(Ch_1, \ldots, Ch_t) = \mathcal{H}\big(M^*, \mathbf{A}, \mathbf{u}, \mathbf{B}^*, \mathbf{b}^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big)$$

is at most $3^{-t}$. Therefore, with probability at least $\epsilon - 3^{-t}$, there exists certain $\kappa^* \le q_{\mathcal{H}}$ such that the $\kappa^*$-th oracle queries involves the tuple

$$\big(M^*, \mathbf{A}, \mathbf{u}, \mathbf{B}^*, \mathbf{b}^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big).$$

Next, $\mathcal{F}$ picks $\kappa^*$ as the target forking point and replays $\mathcal{A}$ many times with the same random tape and input as in the original run. In each rerun, for the first $\kappa^* - 1$ queries, $\mathcal{A}$ is given the same answers $r_1, \ldots, r_{\kappa^*-1}$ as in the initial run, but from the $\kappa^*$-th query onwards, $\mathcal{F}$ replies with fresh random values $r'_{\kappa^*}, \ldots, r'_{q_{\mathcal{H}}} \xleftarrow{\$} \{1,2,3\}^t$. The Improved Forking Lemma of Pointcheval and Vaudenay [44, Lemma 7] implies that, with probability larger than $1/2$, algorithm $\mathcal{F}$ can obtain a 3-fork involving the tuple $\big(M^*, \mathbf{A}, \mathbf{u}, \mathbf{B}^*, \mathbf{b}^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big)$ after less than $32\cdot q_{\mathcal{H}}/(\epsilon - 3^{-t})$ executions of $\mathcal{A}$. Now, let the answers of $\mathcal{F}$ with respect to the 3-fork branches be

$$r_{\kappa^*}^{(1)} = (Ch_1^{(1)}, \ldots, Ch_t^{(1)});\ r_{\kappa^*}^{(2)} = (Ch_1^{(2)}, \ldots, Ch_t^{(2)});\ r_{\kappa^*}^{(3)} = (Ch_1^{(3)}, \ldots, Ch_t^{(3)}).$$

A simple calculation shows that:

$$\Pr\big[\exists i \in \{1, \ldots, t\}:\ \{Ch_i^{(1)}, Ch_i^{(2)}, Ch_i^{(3)}\} = \{1,2,3\}\big] = 1 - (7/9)^t.$$

Conditioned on the existence of such index $i$, one parses the 3 forgeries corresponding to the fork branches to obtain $\big(\mathrm{RSP}_i^{(1)}, \mathrm{RSP}_i^{(2)}, \mathrm{RSP}_i^{(3)}\big)$. They turn out to be 3 *valid* responses with respect to 3 different challenges for the same commitment $\mathrm{CMT}_i$. Since $\mathsf{COM}$ is computationally binding, we can apply Theorem 1 to extract vectors $\mathbf{y} = \big(\mathbf{y}_0\|\mathbf{y}_1^0\|\mathbf{y}_1^1\| \cdots \|\mathbf{y}_\ell^0\|\mathbf{y}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ and $\mathbf{e}^* \in \mathbb{Z}^m$ such that:

1. $\mathbf{y} \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$, and $\mathbf{A}\cdot\mathbf{y} = \mathbf{u} \bmod q$;
2. $\|\mathbf{e}^*\|_\infty \le \beta$ and $\mathbf{b}^* = \mathbf{B}^*\cdot(\mathbf{A}_0\cdot\mathbf{y}_0) + \mathbf{e}^* \bmod q$.

Now we consider two cases:

- If $d \neq d^*$, which happens with probability at most $\frac{N-1}{N}$, then algorithm $\mathcal{F}$ declares $\mathsf{Fail}$ and aborts.
- If $d = d^*$, then let $\mathbf{y}^* = \big(\mathbf{y}_0 \| \mathbf{y}_1^{d^*[1]} \| \ldots \| \mathbf{y}_\ell^{d^*[\ell]}\big) \in \mathbb{Z}^{(\ell+1)m}$, obtained by removing the zero-blocks $\mathbf{y}_1^{1-d^*[1]}, \ldots, \mathbf{y}_\ell^{1-d^*[\ell]}$ from $\mathbf{y}$. Note that, by construction, one has $\mathbf{C} \cdot \mathbf{y}^* = \mathbf{A} \cdot \mathbf{y} = \mathbf{u} = \mathbf{C} \cdot \mathbf{z} \bmod q$.

We will show that, over the randomness of all algorithms, $\mathbf{y}^* \neq \mathbf{z}$ with overwhelming probability. Recall that $\Sigma^*$ is a valid signature such that the implicit tracing algorithm either fails or outputs an index $j^* \notin U \setminus RL^*$.

- If the tracing algorithm fails, then, in particular, one has

  $$\mathsf{Verify}(\mathsf{gpk}, \mathsf{grt}[d^*], \Sigma^*, M^*) = \mathsf{Valid}.$$

  This implies that $\mathbf{A}_0 \cdot \mathbf{y}_0 \neq \mathsf{grt}[d^*] = \mathbf{A}_0 \cdot \mathbf{z}_0 \bmod q$, since otherwise, vector $\mathbf{e}' = \mathbf{b}^* - \mathbf{B}^* \cdot \mathsf{grt}[d^*] = \mathbf{e}^* \bmod q$ would have infinity norm at most $\beta$ and algorithm $\mathsf{Verify}(\mathsf{gpk}, \mathsf{grt}[d^*], \Sigma^*, M^*)$ would output $\mathsf{Invalid}$. As a result, $\mathbf{y}_0 \neq \mathbf{z}_0$, and thus $\mathbf{y}^* \neq \mathbf{z}$.
- If the tracing algorithm outputs $j^* \notin U \setminus RL^*$, namely, the following two facts simultaneously hold true:

  $$\mathsf{Verify}(\mathsf{gpk}, \mathsf{grt}[j^*], \Sigma^*, M^*) = \mathsf{Invalid} \text{ and } \mathsf{Verify}(\mathsf{gpk}, RL^*, \Sigma^*, M^*) = \mathsf{Valid}.$$

  This leads to the following two facts:
  1. $\mathsf{grt}[j^*] \notin RL^*$, and hence, $j^* \notin U$.
  2. Vector $\mathbf{b}^* - \mathbf{B}^* \cdot \mathsf{grt}[j^*] = \mathbf{B}^* \cdot (\mathbf{A}_0 \cdot \mathbf{y}_0 - \mathsf{grt}[j^*]) + \mathbf{e}^* \bmod q$ has infinity norm bounded by $\beta$. Recall that we also have $\|\mathbf{e}^*\|_\infty \leq \beta$. Applying the triangle inequality, we then have

     $$\|\mathbf{B}^* \cdot (\mathbf{A}_0 \cdot \mathbf{y}_0 - \mathsf{grt}[j^*])\|_\infty \leq 2\beta.$$

     By Lemma 4, we deduce that $\mathbf{A}_0 \cdot \mathbf{y}_0 = \mathsf{grt}[j^*] \bmod q$, except for a negligible probability.

  Now we consider two sub-cases:

  1. If $\mathcal{A}$ has never requested the secret key $\mathsf{gsk}[d^*]$, then $\mathbf{z}$ is unknown to $\mathcal{A}$. In this case, because $\mathbf{z}$ has large min-entropy given $\mathbf{u}$ (see Lemma 2), we have $\mathbf{z} \neq \mathbf{y}^*$ with overwhelming probability.
  2. If the adversary $\mathcal{A}$ has requested the secret key $\mathsf{gsk}[d^*]$ in the $\mathsf{Queries}$ phase, then $d^* \in U$. In particular, it must be true that $d^* \neq j^*$ (because $j^* \notin U$), and thus $\mathsf{grt}[d^*] \neq \mathsf{grt}[j^*]$. In other words, we have $\mathbf{A}_0 \cdot \mathbf{y}_0 \neq \mathbf{A}_0 \cdot \mathbf{z}_0 \bmod q$. Therefore, in this case, we also have $\mathbf{y}^* \neq \mathbf{z}$.

Now let $\mathbf{x} = \mathbf{z} - \mathbf{y}^* \in \mathbb{Z}^{(\ell+1)m}$, then we get the following facts: (i) $\mathbf{x} \neq \mathbf{0}$; (ii) $\mathbf{C} \cdot \mathbf{x} = \mathbf{0} \bmod q$; (iii) $\|\mathbf{x}\|_\infty \leq \|\mathbf{z}\|_\infty + \|\mathbf{y}\|_\infty \leq \beta + \beta = 2\beta$. Algorithm $\mathcal{F}$ finally outputs the vector $\mathbf{x}$, which is a valid solution to the given $\mathsf{SIS}^\infty_{n,(\ell+1)\cdot m,q,2\beta}$ instance.

We observe that the probability that $\mathcal{F}$ does not abort is at least $1/N$, and conditioned on not aborting, it can solve the $\mathsf{SIS}^{\infty}_{n,(\ell+1)\cdot m,q,2\beta}$ problem with probability larger than $1/2 \cdot \left(1 - (7/9)^t\right)$ in time

$$T \cdot 32 \cdot q_{\mathcal{H}}/(\epsilon - 3^{-t}) + \mathsf{poly}(n, N).$$

This concludes the proof. □

Acknowledgements. We would like to thank Damien Stehlé, Benoît Libert, Rishiraj Bhattacharyya, Jie Chen, and the anonymous reviewers for their helpful comments. We are grateful to Shota Yamada for pointing out a flaw in the conference version of the paper, and for insightful discussions that lead to the amendment.

The research is supported in part by the Singapore Ministry of Education under Research Grants MOE2013-T2-1-041 and MOE2016-T2-2-014 (S).

## References

1. M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *STOC*, pages 99–108. ACM, 1996.
2. M. Ajtai. Generating Hard Instances of the Short Basis Problem. In *ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
3. J. Alwen and C. Peikert. Generating Shorter Bases for Hard Random Lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011.
4. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
5. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT*, vol. 2656 of LNCS, pages 614–629. Springer, 2003.
6. M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *CT-RSA*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.
7. P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get Shorty via Group Signatures without Encryption. In *SCN*, volume 6280 of *LNCS*, pages 381–398. Springer, 2010.
8. D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *CRYPTO*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
9. D. Boneh and H. Shacham. Group Signatures with Verifier-local Revocation. In *ACM-CCS*, pages 168–177. ACM, 2004.
10. E. Brickell. An Efficient Protocol for Anonymously Providing Assurance of the Container of the Private Key. *Submitted to the Trusted Comp. Group*, April, 2003.
11. J. Camenisch and J. Groth. Group Signatures: Better Efficiency and New Theoretical Aspects. In *SCN*, volume 3352 of *LNCS*, pages 120–133. Springer, 2004.
12. J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.

13. J. Camenisch, G. Neven, and M. Rückert. Fully Anonymous Attribute Tokens from Lattices. In *SCN*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012.
14. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. In *EUROCRYPT*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.
15. D. Chaum and E. van Heyst. Group Signatures. In *EUROCRYPT*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
16. L. Chen and T. P. Pedersen. New Group Signature Schemes (Extended Abstract). In *EUROCRYPT*, volume 950 of *LNCS*, pages 171–181. Springer, 1994.
17. S. Cheng, K. Nguyen, and H. Wang. Policy-Based Signature Scheme from Lattices. *Designs, Codes and Cryptography*, 81(1):43–74, 2016.
18. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
19. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, pages 197–206. ACM, 2008.
20. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A Group Signature Scheme from Lattice Assumptions. In *ASIACRYPT*, volume 6477 of *LNCS*, pages 395–412. Springer, 2010.
21. J. Groth. Fully Anonymous Group Signatures Without Random Oracles. In *ASIACRYPT*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
22. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *ASIACRYPT*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.
23. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-Based Group Signatures with Logarithmic Signature Size. In *ASIACRYPT*, volume 8270 of *LNCS*, pages 41–61. Springer, 2013.
24. A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-Based Group Signature Scheme with Verifier-Local Revocation. In *PKC*, volume 8383 of *LNCS*, pages 345–361. Springer, 2014.
25. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions. In *ASIACRYPT*, volume 10032 of *LNCS*, pages 373–403. Springer, 2016.
26. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors. In *EUROCRYPT*, volume 9666 of *LNCS*, pages 1–31. Springer, 2016.
27. B. Libert, F. Mouhartem, and K. Nguyen. A Lattice-Based Group Signature Scheme with Message-Dependent Opening. In *ACNS*, volume 9696 of *LNCS*, pages 137–155. Springer, 2016.
28. B. Libert, T. Peters, and M. Yung. Group Signatures with Almost-for-Free Revocation. In *CRYPTO*, volume 7417 of *LNCS*, pages 571–589. Springer, 2012.
29. B. Libert and D. Vergnaud. Group Signatures with Verifier-Local Revocation and Backward Unlinkability in the Standard Model. In *CANS*, volume 5888 of *LNCS*, pages 498–517. Springer, 2009.
30. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications. In *PKC*, volume 7778 of *LNCS*, pages 107–124. Springer, 2013.
31. S. Ling, K. Nguyen, and H. Wang. Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based. In *PKC*, volume 9020 of *LNCS*, pages 427–449. Springer, 2015.

32. S. Ling, K. Nguyen, H. Wang, and Y. Xu. Lattice-Based Group Signatures: Achieving Full Dynamicity with Ease. In *ACNS*, volume 10355 of *LNCS*, pages 293–312. Springer, 2017.
33. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
34. D. Micciancio and P. Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484. Springer, 2011.
35. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.
36. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
37. D. Micciancio and O. Regev. Lattice-based Cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.
38. D. Micciancio and S. P. Vadhan. Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2003.
39. T. Nakanishi and N. Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In *ASIACRYPT*, volume 3788 of *LNCS*, pages 533–548. Springer, 2005.
40. T. Nakanishi and N. Funabiki. A Short Verifier-Local Revocation Group Signature Scheme with Backward Unlinkability. In *IWSEC*, volume 4266 of *LNCS*, pages 17–32. Springer, 2006.
41. C. Peikert. Public-key Cryptosystems from the Worst-case Shortest Vector Problem: Extended Abstract. In *STOC*, pages 333–342. ACM, 2009.
42. C. Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In *CRYPTO*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010.
43. C. Peikert and A. Rosen. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In *TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.
44. D. Pointcheval and S. Vaudenay. On Provable Security for Digital Signature Algorithms. *Technical Report LIENS-96-17 of the Laboratoire d'Informatique de Ecole Normale Superieure*, 1997.
45. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC*, pages 84–93. ACM, 2005.
46. M. Rückert. Adaptively Secure Identity-Based Identification from Lattices without Random Oracles. In *SCN*, volume 6280 of *LNCS*, pages 345–362. Springer, 2010.
47. P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
48. J. Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

## A Deferred Analyses of the Interactive Protocol

We first restate Theorem 1.

**Theorem 5.** *Let* $\mathsf{COM}$ *be a statistically hiding and computationally binding string commitment scheme. Then the interactive protocol described in Section 4.1 is a zero-knowledge argument of knowledge with perfect completeness, soundness error* $2/3$*, and communication cost* $\ell \cdot \widetilde{\mathcal{O}}(n)$*. In particular:*

- *There exists an efficient simulator that, on public input* $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{b})$*, 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* $(\mathrm{RSP}^{(1)}, \mathrm{RSP}^{(2)}, \mathrm{RSP}^{(3)})$ *corresponding to all* 3 *possible values of the challenge* $Ch$*, outputs a pair of vector* $(\mathbf{y}, \mathbf{e}') \in \mathbb{Z}^{(2\ell+1)m} \times \mathbb{Z}^m$ *such that:*
  1. $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big) \in \mathsf{Secret}_\beta(d)$ *for some* $d \in \{0,1\}^\ell$*, and* $\mathbf{A} \cdot \mathbf{y} = \mathbf{u} \bmod q$*.*
  2. $\|\mathbf{e}'\|_\infty \leq \beta$ *and* $\mathbf{B} \cdot (\mathbf{A}_0 \cdot \mathbf{y}_0) + \mathbf{e}' = \mathbf{b} \bmod q$*.*

The proof of the theorem is carried out through the next 4 subsections.

### A.1 Completeness and Soundness

An honest prover, given a valid witness $(\mathbf{x}, \mathbf{e}) \in \mathsf{Secret}_\beta(d) \times [-\beta, \beta]^m$, for some $d \in \{0,1\}^\ell$, can always obtain $\mathbf{z}_1, \ldots, \mathbf{z}_p \in \mathsf{SecretExt}(d)$ and $\mathbf{e}_1, \ldots, \mathbf{e}_p \in \mathsf{B}_{3m}$ via the Decomposition-Extension technique. Then if he follows the protocol, he should always get accepted by the verifier. Therefore, the protocol has perfect completeness.

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

### A.2 Communication Cost

The scheme $\mathsf{COM}$ from [22] outputs an element of $\mathbb{Z}_q^n$, and thus, the commitment CMT has bit-size $3n \log q = \widetilde{\mathcal{O}}(n)$. The response RSP is dominated (in size) by $p = \mathcal{O}(\log \beta) = \widetilde{\mathcal{O}}(1)$ permutations in $\mathcal{S}$ and $p$ vectors in $\mathbb{Z}_q^{(2\ell+1)\cdot 3m}$. It can be checked that the bit-size of RSP is $\ell \cdot \widetilde{\mathcal{O}}(n)$. This is also the asymptotic bound for the communication cost of the whole protocol.

### A.3 Zero-Knowledge Property

We will prove that, if $\mathsf{COM}$ is statistically hiding, then the interactive protocol in Section 4.1 is a statistical zero-knowledge argument. Specifically, we construct a PPT simulator $\mathcal{SIM}$ interacting with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$ such that, given only the public input, $\mathcal{SIM}$ outputs with probability close to $2/3$ a simulated transcript that is statistically close to the one produced by the honest prover in the real interaction.

Both simulator $\mathcal{SIM}$ and verifier $\widehat{\mathcal{V}}$ obtain matrices $\mathbf{A}^*, \mathbf{B}^*, \mathbf{I}^*$ from the public input $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{b})$, as specified by the protocol. Then $\mathcal{SIM}$ begins its simulation by selecting a random $\overline{Ch} \in \{1, 2, 3\}$. This is a prediction of the challenge value that $\widehat{\mathcal{V}}$ will *not* choose.

**Case** $\overline{Ch} = 1$: Using linear algebra, $\mathcal{SIM}$ computes the following vectors:

- $\mathbf{z}'_1, \ldots, \mathbf{z}'_p \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}'_j) = \mathbf{u} \bmod q$.
- $\mathbf{z}'_{j,0} = \mathsf{Parse}(\mathbf{z}'_j, 1, m)$, for each $j \in [p]$.
- $\mathbf{e}'_1, \ldots, \mathbf{e}'_p \in \mathbb{Z}_q^{3m}$ such that $\mathbf{B}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}'_{j,0}) + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{e}'_j) = \mathbf{b} \bmod q$.

Next, it samples randomness $\rho_1, \rho_2, \rho_3$ for $\mathsf{COM}$ and

$$\begin{cases} c \xleftarrow{\$} \{0,1\}^\ell; \\ \pi_{z,1}, \ldots, \pi_{z,p} \xleftarrow{\$} \mathcal{S}; \;\; \pi_{e,1}, \ldots, \pi_{e,p} \xleftarrow{\$} \mathcal{S}_{3m}; \\ \mathbf{r}_{z,1}, \ldots, \mathbf{r}_{z,p} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}; \;\; \mathbf{r}_{e,1}, \ldots, \mathbf{r}_{e,p} \xleftarrow{\$} \mathbb{Z}_q^{3m}. \end{cases}$$

For each $j \in [p]$, let $\mathbf{r}_{j,0} = \mathsf{Parse}(\mathbf{r}_{z,j}, 1, m)$. Then $\mathcal{SIM}$ sends $\widehat{\mathcal{V}}$ commitment CMT $= \big(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3\big)$, where

$$\begin{cases} \mathbf{c}'_1 = \mathsf{COM}\big(c,\ \{\pi_{z,j}, \pi_{e,j}\}_{j=1}^p,\ \mathbf{A}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{z,j}); \\ \qquad\qquad \mathbf{B}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}) + \mathbf{I}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{e,j});\ \rho_1\big), \\ \mathbf{c}'_2 = \mathsf{COM}\big(\{\mathrm{T}_c \circ \pi_{z,j}(\mathbf{r}_{z,j}),\ \pi_{e,j}(\mathbf{r}_{e,j})\}_{j=1}^p;\ \rho_2\big), \\ \mathbf{c}'_3 = \mathsf{COM}\big(\{\mathrm{T}_c \circ \pi_{z,j}(\mathbf{z}'_j + \mathbf{r}_{z,j}),\ \pi_{e,j}(\mathbf{e}'_j + \mathbf{r}_{e,j})\}_{j=1}^p;\ \rho_3\big). \end{cases} \tag{7}$$

Receiving a challenge $Ch$ from $\widehat{\mathcal{V}}$, the simulator responds as follows:

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

$$\text{RSP} = \big(c,\ \{\pi_{z,j}, \pi_{e,j}, \mathbf{z}'_j + \mathbf{r}_{z,j}, \mathbf{e}'_j + \mathbf{r}_{e,j}\}_{j=1}^p,\ \rho_1, \rho_3\big). \tag{8}$$

- If $Ch = 3$: Send

$$\text{RSP} = \big(c,\ \{\pi_{z,j}, \pi_{e,j}, \mathbf{r}_{z,j}, \mathbf{r}_{e,j}\}_{j=1}^p,\ \rho_1, \rho_2\big). \tag{9}$$

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

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

For each $j \in [p]$, let $\mathbf{r}_{j,0} = \mathsf{Parse}(\mathbf{r}_{z,j}, 1, m)$. Then $\mathcal{SIM}$ sends the commitment CMT computed in the same manner as in (7).

Receiving a challenge $Ch$ from $\widehat{\mathcal{V}}$, it responds as follows:

- If $Ch = 1$: Send

$$\text{RSP} = \big(d' \oplus c,\ \{\mathrm{T}_c \circ \pi_{z,j}(\mathbf{z}'_j), \mathrm{T}_c \circ \pi_{z,j}(\mathbf{r}_{z,j}), \pi_{e,j}(\mathbf{e}'_j), \pi_{e,j}(\mathbf{r}_{e,j})\}_{j=1}^{p},\ \rho_2, \rho_3\big). \tag{10}$$

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

**Case** $\overline{Ch} = 3$: The simulator proceeds the preparation as in the case $\overline{Ch} = 2$ above. It additionally computes $\mathbf{z}'_{j,0} = \mathsf{Parse}(\mathbf{z}'_j, 1, m)$, for each $j \in [p]$. Then it sends the commitment $\text{CMT} = \big(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3\big)$, where $\mathbf{c}'_2, \mathbf{c}'_3$ are computed as in (7), while

$$\mathbf{c}'_1 = \mathsf{COM}\big(c,\ \{\pi_{z,j}, \pi_{e,j}\}_{j=1}^{p},\ \mathbf{A}^*(\sum_{j=1}^{p}\beta_j \cdot (\mathbf{z}'_j + \mathbf{r}_{z,j})) - \mathbf{u};$$

$$\mathbf{B}^*(\sum_{j=1}^{p}\beta_j \cdot (\mathbf{z}'_{j,0} + \mathbf{r}_{j,0})) + \mathbf{I}^*(\sum_{j=1}^{p}\beta_j \cdot (\mathbf{e}'_j + \mathbf{r}_{e,j})) - \mathbf{b};\ \rho_1\big).$$

Receiving a challenge $Ch$ from $\widehat{\mathcal{V}}$, it 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.

We observe that, in every case we have considered above, since $\mathsf{COM}$ is statistically hiding, the distribution of the commitment CMT and the distribution of the challenge $Ch$ from $\widehat{\mathcal{V}}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly far from 1/3. Moreover, one can check that, whenever the simulator does not halt, it provides an accepting transcript, the distribution of which is statistically close to that of the prover in the real interaction. In other words, we have designed a simulator that can successfully emulate the honest prover with probability negligibly far from 2/3.

### A.4 Argument of Knowledge

We prove that, if $\mathsf{COM}$ is computationally binding, then the given protocol is an argument of knowledge. To this end, it suffices to demonstrate that the protocol has the special soundness property, i.e., given a commitment CMT and 3 valid responses $\text{RSP}^{(1)}$, $\text{RSP}^{(2)}$, $\text{RSP}^{(3)}$ to all 3 possible values of the challenge $Ch$, one can extract a valid witness.

Specifically, let $\text{CMT} = \big(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\big)$, and let $\text{RSP}^{(1)}$, $\text{RSP}^{(2)}$, $\text{RSP}^{(3)}$ as in (3), (4), and (5), respectively. Since all 3 responses satisfy the verification conditions, the following are true:

$$\forall j \in [p]:\ \mathbf{v}_{z,j} \in \mathsf{SecretExt}(d_1); \mathbf{v}_{e,j} \in \mathsf{B}_{3m};$$

$$\mathbf{s}_{j,0} = \mathsf{Parse}(\mathbf{s}_{z,j}, 1, m);\ \mathbf{h}_{j,0} = \mathsf{Parse}(\mathbf{h}_{z,j}, 1, m);$$

$$\mathbf{c}_1 = \mathsf{COM}\big(d_2,\ \{\phi_{z,j}, \phi_{e,j}\}_{j=1}^p,\ \mathbf{A}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{z,j}) - \mathbf{u};$$

$$\mathbf{B}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}) + \mathbf{I}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{e,j}) - \mathbf{b};\ \rho_1\big)$$

$$= \mathsf{COM}\big(d_3,\ \{\psi_{z,j}, \psi_{e,j}\}_{j=1}^p,\ \mathbf{A}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{z,j});$$

$$\mathbf{B}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{j,0}) + \mathbf{I}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{e,j});\ \rho_1\big);$$

$$\mathbf{c}_2 = \mathsf{COM}\big(\{\mathbf{w}_{z,j},\, \mathbf{w}_{e,j}\}_{j=1}^p;\, \rho_2\big) = \mathsf{COM}\big(\{\mathrm{T}_{d_3} \circ \psi_{z,j}(\mathbf{h}_{z,j}),\, \psi_{e,j}(\mathbf{h}_{e,j})\}_{j=1}^p;\, \rho_2\big);$$

$$\mathbf{c}_3 = \mathsf{COM}\big(\{\mathbf{v}_{z,j} + \mathbf{w}_{z,j},\ \mathbf{v}_{e,j} + \mathbf{w}_{e,j}\}_{j=1}^p;\ \rho_3\big)$$
$$= \mathsf{COM}\big(\{\mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}),\ \phi_{e,j}(\mathbf{s}_{e,j})\}_{j=1}^p;\ \rho_3\big).$$

The computational binding property of $\mathsf{COM}$ implies that:

$$\begin{cases} d_2 = d_3; \\ \forall j \in [p] : \phi_{z,j} = \psi_{z,j};\ \mathbf{w}_{z,j} = \mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{h}_{z,j})\ \mathbf{v}_{z,j} + \mathbf{w}_{z,j} = \mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}); \\ \forall j \in [p] : \phi_{e,j} = \psi_{e,j};\ \mathbf{w}_{e,j} = \phi_{e,j}(\mathbf{h}_{e,j}) \text{ and } \mathbf{v}_{e,j} + \mathbf{w}_{e,j} = \phi_{e,j}(\mathbf{s}_{e,j}); \\ \mathbf{A}^*(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_{z,j} - \mathbf{h}_{z,j})) = \mathbf{u} \bmod q; \\ \mathbf{B}^*(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_{j,0} - \mathbf{h}_{j,0})) + \mathbf{I}^*(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_{e,j} - \mathbf{h}_{e,j})) = \mathbf{b} \bmod q. \end{cases}$$

For each $j \in [p]$, let $\mathbf{y}'_j = \mathbf{s}_{z,j} - \mathbf{h}_{z,j}$. Then we have:

$$\mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{y}'_j) = \mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{s}_{z,j}) - \mathrm{T}_{d_2} \circ \phi_{z,j}(\mathbf{h}_{z,j}) = \mathbf{v}_{z,j} \in \mathsf{SecretExt}(d_1).$$

It then follows that $\phi_{z,j}(\mathbf{y}'_j) \in \mathsf{SecretExt}(d_1 \oplus d_2)$. Let $d = d_1 \oplus d_2$, then $\mathbf{y}'_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, since the permutation $\phi_{z,j} \in \mathcal{S}$ preserves the arrangements of the blocks of $\mathbf{y}'_j$.

Now let $\mathbf{y}' = \sum_{j=1}^p \beta_j \cdot \mathbf{y}'_j \in \mathbb{Z}_q^{(2\ell+1)3m}$, and let $\mathbf{y} \in \mathbb{Z}^{(2\ell+1)m}$ be the vector obtained from $\mathbf{y}'$ by removing the last $2m$ coordinates in each $3m$-block. We note that

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

Moreover, as $\mathbf{y}'_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, we have that $\mathbf{y} \in \mathsf{Secret}_\beta(d)$. Furthermore,

$$\mathbf{A} \cdot \mathbf{y} = \mathbf{A}^* \cdot \mathbf{y}' = \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{y}_j = \mathbf{A}^*(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_{z,j} - \mathbf{h}_{z,j})) = \mathbf{u} \bmod q.$$

Write vector $\mathbf{y}$ as $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big)$, then observe that $\mathbf{y}_0 = \sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{j,0} - \mathbf{h}_{j,0})$.

On the other hand, for each $j \in [p]$, let $\mathbf{e}'_j = \mathbf{s}_{e,j} - \mathbf{h}_{e,j}$. Then we have,

$$\phi_{e,j}(\mathbf{e}'_j) = \phi_{e,j}(\mathbf{s}_{e,j}) - \phi_{e,j}(\mathbf{h}_{e,j}) = \mathbf{v}_{e,j} \in \mathsf{B}_{3m},$$

which implies that $\mathbf{e}'_j \in \mathsf{B}_{3m}$. Let $\widehat{\mathbf{e}} = \sum_{j=1}^{p} \beta_j \cdot \mathbf{e}'_j \in \mathbb{Z}^{3m}$ and let $\mathbf{e}' \in \mathbb{Z}^m$ be the vector obtained by dropping the last $2m$ coordinates from $\widehat{\mathbf{e}}$. We then note that:

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

Observe further that we have the relation:

$$\mathbf{B}^* \cdot \mathbf{y}_0 + \mathbf{I}^* \cdot \widehat{\mathbf{e}} = \mathbf{b} \bmod q \iff \mathbf{B} \cdot (\mathbf{A}_0 \cdot \mathbf{y}_0) + \mathbf{e}' = \mathbf{b} \bmod q.$$

In conclusion, we have constructed an efficient knowledge extractor that outputs a pair of vectors $(\mathbf{y}, \mathbf{e}')$ satisfying all the conditions stated in Theorem 1. □