Lattice-based Group Signature Scheme with Verifier-local Revocation

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

[1] École Normale Supérieure de Lyon,
LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL),
46 Allée d'Italie, 69364 Lyon Cedex 07, France.
adeline.langlois@ens-lyon.fr
[2] Division of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore.
{lingsan, khoantt, hxwang}@ntu.edu.sg

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 weaker security assumption. In the random oracle model, our scheme is proved to be secure based on the hardness of the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$ problem in general lattices - an assumption that is as weak as those of state-of-the-art lattice-based standard signatures. Moreover, our construction works without relying on encryption schemes, which is an intriguing feature for group signatures.

Keywords: group signature, verifier-local revocation, lattice-based cryptography

1 Introduction

Group Signatures. 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 ([16,4,5,8,9,6,20,23], ...).

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 [31,32], Libert and Vergnaud [24], 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 [38]. 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 ([1,36,18,29]). 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. [19] 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. [22] 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 encryption based on the Learning With Errors (LWE) problem, and then generate a sophisticated proof to prove particularly that the ciphertext is well-formed. Relying on encryption 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 encryption scheme is secure, it usually leads to a relatively strong security assumption. In particular, the recent scheme by Laguillaumie et al. [22] is only provably secure if there is no quantum algorithm to approximate the Shortest Independent Vectors Problem (SIVP_γ) 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 encryptions? How to design a more efficient scheme based on weaker security assumption?

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 [19], [13] and [22] 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. The signature is basically an all-in-one proof of knowledge, made non-interactive using Fiat-Shamir paradigm [17]. Moreover, the scheme departs from the traditional paradigm, and is free of LWE-based 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 result is comparable to that of [22], and is a noticeable improvement over those of [19] 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 $\gamma = \widetilde{\mathcal{O}}(n^{1.5})$. Surprisingly, this security assumption is as weak as those of state-of-the-art lattice-based *standard* signatures, such as [18], [14], and [27]. This is a non-trivial feature, because when constructing group signatures, which is a more eleborate primitive than standard signatures, one would expect to rely on a stronger security assumption.

Overview of Our Techniques. The main building block of our VLR group signature scheme is an interactive protocol allowing a prover to convince the verifier that he is a certified group member (i.e., he possesses a valid secret signing key), and that he has not been revoked (i.e., his "revocation token" is not in the verifier's blacklist). The protocol is repeated many times to make the soundness error negligibly small, and then is converted to a signature scheme via Fiat-Shamir heuristic. Roughly speaking, in the random oracle model, the traceability and anonymity of the resulting group signature are based on the facts that the underlying protocol is a proof of knowledge, and it can be simulated.

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, β, 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], namely, 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 of 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 (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 of knowledge (e.g., [30,26,25]) 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 [37]), but it is insufficient for our purposes: to achieve anonymity, the group user also 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 ℓ suitable *zero-blocks* of size m to vector $\mathbf{z}$ to obtain an 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 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' proof system from [25], 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 briefly review 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 $\mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q \in \mathbb{Z}_q^n$. We choose suitable parameters, and sample $\mathbf{x}_0$ from a proper distribution, so that the token is statistically close to uniform over $\mathbb{Z}_q^n$. At a high level, our revocation mechanism works as follows. The user is asked to sample a uniformly random vector $\mathbf{r}_0 \in \mathbb{Z}_q^m$, and to compute a commitment $\mathbf{c}_0$ using a (lattice-based) statistically hiding and computationally binding string commitment scheme COM, for which the value $\mathbf{A}_0 \cdot \mathbf{r}_0 \bmod q$ is part of the committed string. Depending on the verifier's challenge, the user will either reveal $\mathbf{r}_0$ or reveal $\mathbf{x}_0 + \mathbf{r}_0$. In the former case, the verifier can check for honest computation of $\mathbf{c}_0$, while in the latter case, he can perform the revocation check using a list of tokens of revoked users $RL = \big\{\{\mathbf{u}_i\}_i\big\} \subset \mathbb{Z}_q^n$, as follows:

$$\forall \mathbf{u}_i \in RL, \text{ check that } \mathbf{c}_0 \neq \mathsf{COM}\big(\mathbf{A}_0 \cdot (\mathbf{x}_0 + \mathbf{r}_0) - \mathbf{u}_i \bmod q\big).$$

Assuming that the user has been revoked, i.e., there exists i such that $\mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q = \mathbf{u}_i$. If he follows the protocol, then $\mathsf{COM}\big(\mathbf{A}_0 \cdot (\mathbf{x}_0 + \mathbf{r}_0) - \mathbf{u}_i \bmod q\big) = \mathsf{COM}(\mathbf{A}_0 \cdot \mathbf{r}_0 \bmod q) = \mathbf{c}_0$, and thus, he gets rejected. If there is a false acceptance, then we can use it to break the computational binding property of COM. On the other hand, the probability of false rejection is negligibly small, since COM is statistically regular.

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" proof system [25]. We also employ a special *"one-time pad"* technique, and a novel revocation mechanism.

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 $1 \le i_1 \le i_2 \le n$. 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 $\varDelta(D_1, D_2) = \frac{1}{2}\sum_{x\in S}|D_1(x) - D_2(x)|$. Two distributions are statistically close if their statistical distance is negligible.

2.1 VLR Group Signature

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

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

Remark 1. Any VLR group signature has an *implicit tracing algorithm* using grt as the tracing key. The tracing algorithm works as follows: on input a valid signature Σ on a message M, it reveals the signer of Σ 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 $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$ outputted by KeyGen, 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 KeyGen to generate $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$, then gives gpk to the adversary $\mathcal{A}$.
(b) $\mathsf{Queries}$. 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) $\mathsf{Challenge}$. 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 $b \xleftarrow{\$} \{0,1\}$, computes a signature of user d_b on M^* as $\Sigma^* = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d_b], M^*)$, and returns Σ^* 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) Output. Eventually, $\mathcal{A}$ outputs a bit b'. It wins the game if $b' = b$.
We define the adversary's advantage in winning the game as $\mathsf{Adv}_{\mathcal{A}} = \big|\Pr[b' = b] - 1/2\big|$. 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) 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 $\Sigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$.
   - 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 Σ^*. The adversary wins the game if:
   i. $\mathsf{Verify}(\mathsf{gpk}, RL^*, \Sigma^*, M^*) = \mathsf{valid}$.
   ii. The (implicit) tracing algorithm fails or traces to a user outside of the coalition $U \setminus RL^*$.
   iii. The signature Σ^* is non-trivial, i.e., $\mathcal{A}$ did not obtain Σ^* 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 KeyGen and 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:

$$\Lambda^{\perp}(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^m : \ \mathbf{A}\cdot\mathbf{x} = \mathbf{0} \bmod q\right\} \subseteq \mathbb{Z}^m.$$

For any $\mathbf{u}$ in the image of $\mathbf{A}$, define the coset $\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^m : \ \mathbf{A}\cdot\mathbf{x} = \mathbf{u} \bmod q\right\}$. We recall the homogeneous and inhomogeneous Small Integer Solution problems (SIS and ISIS).

Definition 1. *The $\mathsf{SIS}^p_{n,m,q,\beta}$ and $\mathsf{ISIS}^p_{n,m,q,\beta}$ problem in the ℓ_p norm with parameters (n,m,q,β) 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 $\mathbf{x} \in \Lambda^{\perp}(\mathbf{A})$ such that $\|\mathbf{x}\|_p \leq \beta$.*
- $\mathsf{ISIS}^p_{n,m,q,\beta}$ *asks to find a vector $\mathbf{x} \in \Lambda_{\mathbf{u}}^{\perp}(\mathbf{A})$ such that $\|\mathbf{x}\|_p \leq \beta$.*

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

Theorem 1 ([18]). *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 SIVP^2_{γ} 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 ℓ_2 and ℓ_∞ 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 SIVP^2_{γ} (in the ℓ_2 norm) for some $\gamma = \beta\cdot\widetilde{O}(n)$.

Gaussians over Lattices. For any positive real σ, 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 Λ, define the discrete Gaussian distribution over Λ 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 1 ([18][34]). *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 $D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}),\sigma}$.*
2. *For $\beta = \lceil\sigma\cdot\log m\rceil$, and $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, $\Pr\left[\|\mathbf{x}\|_\infty > \beta\right]$ 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 theorem are improvements of those in the literature [2,18,33,3].

Theorem 2 ([28]). *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 $\mathbf{R}_{\mathbf{A}}$, 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* $\mathsf{SampleD}(\mathbf{R}_{\mathbf{A}},\mathbf{A},\mathbf{u},\sigma)$ *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. [21] constructed a string commitment scheme COM : $\{0,1\}^* \times \{0,1\}^{\overline{m}/2} \to \mathbb{Z}_q^n$, such that:

- If $\overline{m} > 2n(1+\delta)\log q$ for some positive constant δ, 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 use the KTX commitment scheme. For simplicity, we will omit the randomness of the commitment. Also, we implicitly choose $\overline{m}$ sufficiently large, e.g., $\overline{m} = 4n\log q$, to make COM statistically hiding.

3 Preparations

We now describe the parameters and some specific constructions 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**
Modulus q	$\omega(n^2 \log n)$
Dimension m	$\geq 2n\log q$
Gaussian parameter σ	$\omega(\sqrt{n \log q \log n})$
Integer norm bound β	$\lceil \sigma \cdot \log m \rceil$
Number of 'decompositions' p	$\lfloor \log \beta \rfloor + 1$
Sequence of integers $\beta_1, \beta_2, \beta_3, \ldots, \beta_p$	$\beta_1 = \lceil \beta/2 \rceil; \beta_2 = \lceil (\beta - \beta_1)/2 \rceil$ $\beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil; \ldots; \beta_p = 1$
Number of protocol repetitions t	$\omega(\log n)$

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 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 two sets:

- $\mathsf{Secret}_{\beta}(d)$: The set of all vectors $\mathbf{x} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_{\ell}^0 \| \mathbf{x}_{\ell}^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ consisting of $2\ell+1$ *blocks* of size m, such that $\|\mathbf{x}\|_{\infty} \leq \beta$, and the following ℓ 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 vectors $\mathbf{x} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_{\ell}^0 \| \mathbf{x}_{\ell}^1\big) \in \{-1,0,1\}^{(2\ell+1)3m}$ 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 B_{3m}, and the remaining ℓ 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 $\tau_0, \tau_1^0, \tau_1^1, \ldots, \tau_\ell^0, \tau_\ell^1$ are certain 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 $d, e \in \{0,1\}^\ell$, $\pi \in \mathcal{S}$, and $\mathbf{x} \in \mathbb{Z}^{(2\ell+1)3m}$, it can be checked that:
$$\mathbf{x} \in \mathsf{SecretExt}(d) \Leftrightarrow \pi(\mathbf{x}) \in \mathsf{SecretExt}(d) \Leftrightarrow T_e \circ \pi(\mathbf{x}) \in \mathsf{SecretExt}(d \oplus e). \tag{1}$$

3.3 The Decomposition - Extension Technique

Ling et al. [25] 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 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 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 [25] 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 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 $\mathbf{w} = \big(\widetilde{\mathbf{w}}\|\widehat{\mathbf{w}}\big) \in \mathsf{B}_{3m}$.

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 $\mathbf{x}$ 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 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 EleExt on each of the decomposed vectors to obtain $(\ell+1)p$ vectors in 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 ℓp *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 $\mathbf{A}^* \in \mathbb{Z}_q^{n\times(2\ell+1)3m}$:

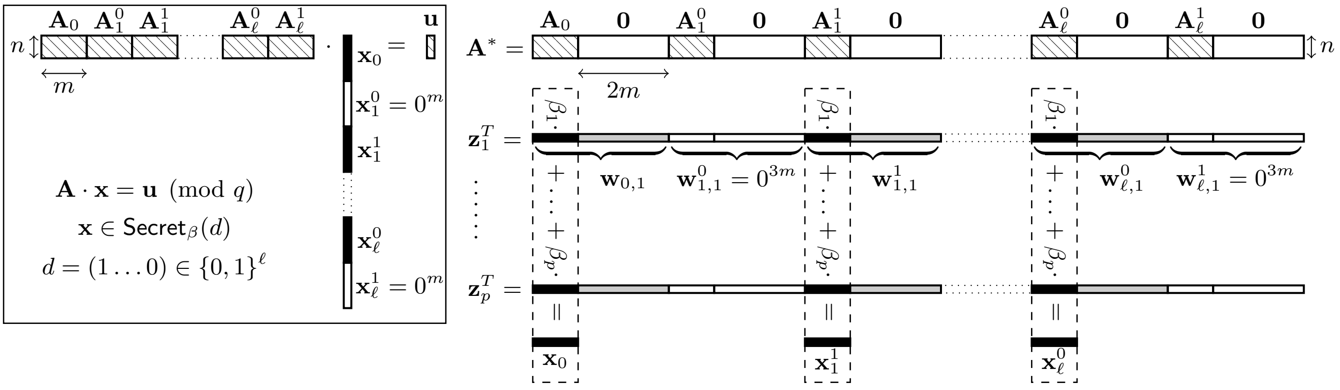
$\mathbf{A}_0$ $\mathbf{A}_1^0$ $\mathbf{A}_1^1$ $\mathbf{A}_\ell^0$ $\mathbf{A}_\ell^1$
$\mathbf{u}$
n
m
$\mathbf{x}_0 =$
$\mathbf{x}_1^0 = 0^m$
$\mathbf{x}_1^1$
$\mathbf{x}_\ell^0$
$\mathbf{x}_\ell^1 = 0^m$
$\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \pmod q$
$\mathbf{x} \in \mathsf{Secret}_\beta(d)$
$d = (1 \dots 0) \in \{0,1\}^\ell$
$\mathbf{A}^* =$
$\mathbf{A}_0$ $\mathbf{0}$ $\mathbf{A}_1^0$ $\mathbf{0}$ $\mathbf{A}_1^1$ $\mathbf{0}$ $\mathbf{A}_\ell^0$ $\mathbf{0}$ $\mathbf{A}_\ell^1$ $\mathbf{0}$
$2m$
$\mathbf{z}_1^T =$
$\mathbf{z}_p^T =$
$\beta_1 \cdot$
$+ \dots +$
$\beta_p \cdot$
$\mathbf{w}_{0,1}$
$\mathbf{w}_{1,1}^0 = 0^{3m}$
$\mathbf{w}_{1,1}^1$
$\mathbf{w}_{\ell,1}^0$
$\mathbf{w}_{\ell,1}^1 = 0^{3m}$
$\mathbf{x}_0$
$\mathbf{x}_1^1$
$\mathbf{x}_\ell^0$

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 B_{3m}. After performing Decomposition-Extension, one has that $\mathbf{z}_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, and $\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j\big) = \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.

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:

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

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

4 The Underlying Interactive Protocol

We recall that the main building block of our VLR group signature scheme is an interactive protocol that allows the prover to convince the verifier that he is a certified group member (i.e., he has a valid secret key), and that he has not been revoked (i.e., his revocation token is not in the verifier's list RL). 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 parameters are $\mathbf{A} = \big[\mathbf{A}_0 \big| \mathbf{A}_1^0 \big| \mathbf{A}_1^1 \big| \dots \big| \mathbf{A}_\ell^0 \big| \mathbf{A}_\ell^1\big] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$.
- The prover's witness is a $\mathbf{x} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big) \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$. The verifier's additional input is a set $RL = \big\{\{\mathbf{u}_i\}_i\big\} \subset \mathbb{Z}_q^n$, whose cardinality is at most $N - 1$.
- The prover's goal is to convince the verifier in that:
  1. $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$ and $\mathbf{x} \in \mathsf{Secret}_\beta(d)$, while keeping d *secret.*
  2. $\mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q \notin RL$.

4.1 Description of the Protocol

Let COM be the KTX commitment scheme [21]. Let $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$. Prior to the interaction, the prover applies the Decomposition-Extension technique on his witness: Let $\mathbf{z}_1, \dots, \mathbf{z}_p \leftarrow \mathsf{WitnessDE}(\mathbf{x})$.

The protocol follows Stern's approach for three-pass zero-knowledge identification schemes [39], for which we employ an additional commitment $\mathbf{c}_0$ to enable the revocation mechanism. The details are as follows:

1. **Commitment:** The prover samples a string $e \xleftarrow{\$} \{0,1\}^\ell$, p permutations $\pi_1, \ldots, \pi_p \xleftarrow{\$} \mathcal{S}$, and p vectors $\mathbf{r}_1, \ldots, \mathbf{r}_p \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}$. For each $j \in [p]$, let $\mathbf{r}_{j,0} = \mathsf{Parse}(\mathbf{r}_j, 1, m)$. Then it sends the commitment $\mathrm{CMT} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in (\mathbb{Z}_q^n)^4$ to the verifier, where
$$\begin{cases} \mathbf{c}_0 = \mathsf{COM}\big(e,\ \{\pi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big) \bmod q\big), \\ \mathbf{c}_1 = \mathsf{COM}\big(e,\ \{\pi_j\}_{j=1}^p,\ \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j\big) \bmod q\big), \\ \mathbf{c}_2 = \mathsf{COM}\big(\{\mathrm{T}_e \circ \pi_j(\mathbf{r}_j)\}_{j=1}^p\big), \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathrm{T}_e \circ \pi_j(\mathbf{z}_j + \mathbf{r}_j)\}_{j=1}^p\big). \end{cases} \tag{2}$$
2. **Challenge:** The verifier sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$ to the prover.
3. **Response:** Depending on the challenge, the prover computes the response RSP differently:
   - Case $Ch = 1$: $\forall\, j \in [p]$, let $\mathbf{v}_j = \mathrm{T}_e \circ \pi_j(\mathbf{z}_j)$, $\mathbf{w}_j = \mathrm{T}_e \circ \pi_j(\mathbf{r}_j)$, $d_1 = d \oplus e$, and set:
   $$\mathrm{RSP} = \big(d_1,\ \{\mathbf{v}_j\}_{j=1}^p,\ \{\mathbf{w}_j\}_{j=1}^p\big). \tag{3}$$
   - Case $Ch = 2$: $\forall\, j \in [p]$, let $\phi_j = \pi_j$, $\mathbf{s}_j = \mathbf{z}_j + \mathbf{r}_j$, $d_2 = e$, and set:
   $$\mathrm{RSP} = \big(d_2,\ \{\phi_j\}_{j=1}^p,\ \{\mathbf{s}_j\}_{j=1}^p\big). \tag{4}$$
   - Case $Ch = 3$: $\forall\, j \in [p]$, let $\psi_j = \pi_j$, $\mathbf{h}_j = \mathbf{r}_j$, $d_3 = e$, and set:
   $$\mathrm{RSP} = \big(d_3,\ \{\psi_j\}_{j=1}^p,\ \{\mathbf{h}_j\}_{j=1}^p\big). \tag{5}$$

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

- Case $Ch = 1$: Parse RSP as in (3). Check that $\forall j \in [p] :\ \mathbf{v}_j \in \mathsf{SecretExt}(d_1)$, and that:
$$\mathbf{c}_2 = \mathsf{COM}\big(\{\mathbf{w}_j\}_{j=1}^p\big) \text{ and } \mathbf{c}_3 = \mathsf{COM}\big(\{\mathbf{v}_j + \mathbf{w}_j\}_{j=1}^p\big).$$
- Case $Ch = 2$: Parse RSP as in (4). $\forall j \in [p]$, let $\mathbf{s}_{j,0} = \mathsf{Parse}(\mathbf{s}_j, 1, m)$. Check that:
$$\begin{cases} \forall \mathbf{u}_i \in RL:\ \mathbf{c}_0 \neq \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big) - \mathbf{u}_i \bmod q\big) \\ \mathbf{c}_1 = \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j\big) - \mathbf{u} \bmod q\big);\ \ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j)\}_{j=1}^p\big). \end{cases}$$
- Case $Ch = 3$: Parse RSP as in (5). $\forall j \in [p]$, let $\mathbf{h}_{j,0} = \mathsf{Parse}(\mathbf{h}_j, 1, m)$. Check that:
$$\begin{cases} \mathbf{c}_0 = \mathsf{COM}(d_3,\ \{\psi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{j,0}) \bmod q\big) \\ \mathbf{c}_1 = \mathsf{COM}\big(d_3,\ \{\psi_j\}_{j=1}^p,\ \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{h}_j) \bmod q\big);\ \ \mathbf{c}_2 = \mathsf{COM}\big(\{\mathrm{T}_{d_3} \circ \psi_j(\mathbf{h}_j)\}_{j=1}^p\big). \end{cases}$$

The verifier outputs Valid if and only if all the conditions hold. Otherwise, he outputs $\mathsf{Invalid}$.

4.2 Witness Extraction

The following lemma says that in our protocol, one can extract a satisfying witness under specific conditions.

Lemma 2. *Assume that for a given commitment* CMT, *there exist* 3 valid *responses* $\mathrm{RSP}^{(1)}$, $\mathrm{RSP}^{(2)}$, *and* $\mathrm{RSP}^{(3)}$ *corresponding to all* 3 *possible values of the challenge* Ch. *If* COM *is a computationally binding commitment scheme, then one can efficiently extract a vector* $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \cdots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ *satisfying* $\mathbf{A} \cdot \mathbf{y} = \mathbf{u} \bmod q$, $\mathbf{y} \in \mathsf{Secret}_\beta(d)$ *for some* $d \in \{0,1\}^\ell$, *and* $\mathbf{A}_0 \cdot \mathbf{y}_0 \bmod q \notin RL$.

The proof of this lemma 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.

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 vectors $\mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$. Compute $\mathbf{z} = \sum_{i=1}^{\ell} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]} \bmod q$, and sample $\mathbf{x}_0 \in \mathbb{Z}^m$ with $\mathbf{x}_0 \hookleftarrow \mathsf{SampleD}\big(\mathbf{R}, \mathbf{A}_0, \mathbf{u}-\mathbf{z}, \sigma\big)$. 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 Theorem 2, 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 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 Theorem 2 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 1). 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 1, 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 $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$ be a hash function, modelled as a random oracle. Given $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, to sign a message $M \in \{0,1\}^*$ using the secret key $\mathsf{gsk}[d] = \mathbf{x} \in \mathsf{Secret}_\beta(d)$, the user runs the randomized algorithm $\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$, which performs the following steps:

1. Generate a proof that the user is a certified group members and that he has not been revoked. This is done by repeating $t = \omega(\log n)$ times the basic protocol from Section 4 with public parameter $(\mathbf{A}, \mathbf{u})$ and prover's witness $\mathbf{x}$, and then making it non-interactive with the Fiat-Shamir heuristic as a triple $\big(\{\mathrm{CMT}^{(k)}\}_{k=1}^t,\ \mathrm{CH},\ \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$, where
$$\mathrm{CH} = \big(\{Ch^{(k)}\}_{k=1}^t\big) = \mathcal{H}\big(M,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big) \in \{1,2,3\}^t.$$
2. Output the group signature:
$$\varSigma = \big(M,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t,\ \{Ch^{(k)}\}_{k=1}^t,\ \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big). \tag{6}$$

Verification Algorithm. On input $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, 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 $\varSigma$ on M, the verifier runs the deterministic algorithm $\mathsf{Verify}(\mathsf{gpk}, RL, \varSigma, M)$, which performs the following steps:

1. Parse the signature $\varSigma$ as in (6).
2. Check if $\big(Ch^{(1)}, \ldots, Ch^{(t)}\big) = \mathcal{H}\big(M, \mathrm{CMT}^{(1)}, \ldots, \mathrm{CMT}^{(t)}\big)$.
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 output $\mathsf{Invalid}$ and terminate.
4. Output 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. Asymptotically, 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, i.e., the check against $\mathbf{c}_0^{(k)}$ in the case $Ch^{(k)} = 2$, runs in linear time in the number of revoked users, as it seems unavoidable for secure VLR group signature schemes.

Theorem 3. *Our VLR group signature scheme is correct with overwhelming probability.*

The proof of this Theorem is provided in Appendix B.1.

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

Theorem 4. *If* COM *is a statistically hiding string commitment scheme, then the VLR group signature scheme in Section 5.1 is selfless-anonymous in the random oracle model.*

Proof. We define two hybrid games G_0 and G_1. Game G_0 is the original selfless-anonymity game (see Section 2). In game G_1, we make the distribution of the challenger's output independent of the bit $b \in \{0,1\}$. We then prove that these two games are statistically indistinguishable. Since the adversary's advantage in game G_1 is 0, this implies the selfless-anonymity of our scheme.

Game G_0:

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 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 $d_b \notin \mathsf{Corrupted}$ and $\mathsf{grt}[d_b] \notin RL$ for each $b \in \{0, 1\}$.
4. Pick a bit $b \xleftarrow{\$} \{0, 1\}$, generate a valid signature
$$\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d_b], M^*\big) = \big(M^*,\; \{\mathrm{CMT}^{(k)}\}_{k=1}^{t}, \{Ch^{(k)}\}_{k=1}^{t}, \{\mathrm{RSP}^{(k)}\}_{k=1}^{t}\big),$$
and return Σ to $\mathcal{A}$.
5. $\mathcal{A}$ can still make queries as before, but it is not allowed to ask for $\mathsf{gsk}[d_b]$ or $\mathsf{grt}[d_b]$, for each $b \in \{0, 1\}$.
6. Finally $\mathcal{A}$ outputs a bit b'.

Game G_1:
In this game, we make the following modification with respect to **Game** G_0: In Step 4, instead of generating a legitimate signature, we simulate the signature generation. Our simulation algorithm is such that:

- **Input:** The group public key $\mathsf{gpk} = \big(\mathbf{A}, \mathbf{u}\big)$ obtained from Step 1, the set of user revocation tokens RL obtained at the end of Step 2, and the message M^* obtained from Step 3.
- **Output:** A *valid* group signature Σ^* for message M^* under gpk and RL. Moreover, Σ^* is *independent* of the bit b, and it is statistically indistinguishable from the legitimate signature Σ in game G_0.

Let $\mathbf{A} = \big[\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1\big]$ and $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$. The simulation algorithm does the following:

1. For each $k \in [t]$, pick a "fake" challenge $\overline{Ch}^{(k)} \xleftarrow{\$} \{1, 2, 3\}$, that is a "prediction" of what the real challenge will *not* be. Then pick a real challenge $Ch^{(k)} \xleftarrow{\$} \{1, 2, 3\} \setminus \{\overline{Ch}^{(k)}\}$. It turns out that $Ch^{(k)}$ is uniformly distributed in $\{1, 2, 3\}$, which satisfies the requirement on the output of the random oracle $\mathcal{H}$. Then prepare $\mathrm{CMT}^{(k)}$, and the response $\mathrm{RSP}^{(k)}$ to $(\mathrm{CMT}^{(k)}, Ch^{(k)})$ as follows:
   (a) **Case** $\overline{Ch}^{(k)} = 1$:
      i. Use linear algebra to compute $\mathbf{z} \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\mathbf{A}^* \cdot \mathbf{z} = \mathbf{u} \bmod q$. Let $\mathbf{g}_0 = \mathsf{Parse}(\mathbf{z}, 1, m)$. If $\mathbf{A}_0 \cdot \mathbf{g}_0 \in RL$ then repeat this step. Otherwise, compute $\mathbf{z}_1^{(k)}, \ldots, \mathbf{z}_p^{(k)} \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j^{(k)} = \mathbf{z} \bmod q$.
      ii. Sample $e^{(k)} \xleftarrow{\$} \{0, 1\}^\ell$, and for all $j \in [p]$, sample $\pi_j^{(k)} \xleftarrow{\$} \mathcal{S}$ and $\mathbf{r}_j^{(k)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}$, and let $\mathbf{r}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{r}_j^{(k)}, 1, m)$.
      iii. Compute $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big) \in (\mathbb{Z}_q^n)^4$ as in (2), from Section 4.
      iv. If $Ch^{(k)} = 2$, then set $\mathrm{RSP}^{(k)} = \big(e^{(k)},\; \{\pi_j^{(k)}\}_{j=1}^{p},\; \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^{p}\big)$. (7)
      If $Ch^{(k)} = 3$, then set $\mathrm{RSP}^{(k)} = \big(e^{(k)},\; \{\pi_j^{(k)}\}_{j=1}^{p},\; \{\mathbf{r}_j^{(k)}\}_{j=1}^{p}\big)$. (8)
   (b) **Case** $\overline{Ch}^{(k)} = 2$:

i. Sample $d^{(k)}, e^{(k)} \xleftarrow{\$} \{0,1\}^{\ell}$. For all $j \in [p]$, sample $\pi_j^{(k)} \xleftarrow{\$} \mathcal{S}$, and $\mathbf{r}_j^{(k)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}$, and $\mathbf{z}_j^{(k)} \xleftarrow{\$} \mathsf{SecretExt}(d^{(k)})$. Let $\mathbf{r}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{r}_j^{(k)}, 1, m)$.

ii. Compute $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big) \in (\mathbb{Z}_q^n)^4$ as in (2), from Section 4.

iii. If $Ch^{(k)} = 1$, then set $\mathrm{RSP}^{(k)} = \big(d^{(k)} \oplus e^{(k)},\ \{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})\}_{j=1}^{p},\ \{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\}_{j=1}^{p}\big)$. (9)

If $Ch^{(k)} = 3$, then set $\mathrm{RSP}^{(k)} = \big(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^{p},\ \{\mathbf{r}_j^{(k)}\}_{j=1}^{p}\big)$. (10)

(c) **Case** $\overline{Ch}^{(k)} = 3$:

i. Sample $d^{(k)}, e^{(k)} \xleftarrow{\$} \{0,1\}^{\ell}$. For all $j \in [p]$ sample $\pi_j^{(k)} \xleftarrow{\$} \mathcal{S}$ and $\mathbf{r}_j^{(k)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}$, and let $\mathbf{r}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{r}_j^{(k)}, 1, m)$.

ii. For all $j \in [p]$, sample $\mathbf{z}_j^{(k)} \xleftarrow{\$} \mathsf{SecretExt}(d^{(k)})$, and let $\mathbf{z}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{z}_j^{(k)}, 1, m)$. If $\mathbf{A}_0 \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_{j,0}^{(k)}) \in RL$, then repeat this step.

iii. Compute $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big) \in (\mathbb{Z}_q^n)^4$, where $\mathbf{c}_0^{(k)}, \mathbf{c}_2^{(k)}$ and $\mathbf{c}_3^{(k)}$ are as in (2), from Section 4, while $\mathbf{c}_1^{(k)}$ is computed as follows:

$$\mathbf{c}_1^{(k)} = \mathsf{COM}\Big(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^{p},\ \mathbf{A}^* \cdot \sum_{j=1}^{p} \beta_j \cdot (\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}) - \mathbf{u}\Big).$$

iv. If $Ch^{(k)} = 1$, then set $\mathrm{RSP}^{(k)} = \big(d^{(k)} \oplus e^{(k)},\ \{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})\}_{j=1}^{p},\ \{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\}_{j=1}^{p}\big)$. (11)

If $Ch^{(k)} = 2$, then set $\mathrm{RSP}^{(k)} = \big(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^{p},\ \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^{p}\big)$. (12)

2. Program the random oracle: $\mathcal{H}\big(M^*, \mathrm{CMT}^{(1)}, \ldots, \mathrm{CMT}^{(t)}\big) = \big(Ch^{(1)}, \ldots, Ch^{(t)}\big)$.
3. Output the simulated signature $\Sigma^* = \big(M^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^{t}, \{Ch^{(k)}\}_{k=1}^{t}, \{\mathrm{RSP}^{(k)}\}_{k=1}^{t}\big)$.

We have the following observations on the above construction:

- For every $k \in [t]$, the distribution of $\mathrm{CMT}^{(k)}$ is statistically close to uniform over $(\mathbb{Z}_q^n)^4$. This follows from the statistically hiding property of COM.
- The distribution of $\big(Ch^{(1)}, \ldots, Ch^{(t)}\big)$ is uniform over $\{1,2,3\}^t$.
- For every $k \in [t]$:
  1. If $Ch^{(k)} = 1$, the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)}$ *and* $\mathrm{RSP}^{(k)}$ is either (1(b)ii) *and* (9), or (1(c)iii) *and* (11).
  2. If $Ch^{(k)} = 2$, the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)}$ *and* $\mathrm{RSP}^{(k)}$ is either (1(a)iii) *and* (7), or (1(c)iii) *and* (12).
  3. If $Ch^{(k)} = 3$, the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)}$ *and* $\mathrm{RSP}^{(k)}$ is either (1(a)iii) *and* (8), or (1(b)ii) *and* (10).

  We remark that, in every case, $\mathrm{RSP}^{(k)}$ is intentionally designed to be a valid "response" to $\mathrm{CMT}^{(k)}$ and $Ch^{(k)}$, and to be statistically close to that produced by Step (4) in Game G_0.

These observations imply that Σ^* is a valid group signature, i.e., $\mathsf{Verify}\big((\mathbf{A}, \mathbf{u}), RL, \Sigma^*, M^*\big) = \mathsf{Valid}$, and that Σ^* is statistically indistinguishable from the legitimate signature Σ produced by Game G_0 (for a more detailed analysis, see Lemma 4 in Appendix B.2). It then follows that Game G_0 and Game G_1 are statistically indistinguishable. Moreover, Σ^* is independent of the bit $b \in \{0,1\}$, thus, the adversary's advantage in Game G_1 is 0. As a result, the adversary's advantage in Game G_0 is negligible. In other words, our VLR group signature is selfless-anonymous. □

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 5. *If there is a traceability adversary $\mathcal{A}$ with success probability ϵ 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 Theorem 1 and Theorem 5 imply that our scheme is traceable in the random oracle model, based on the worst-case hardness of the SIVP_γ problem (in the ℓ_2 norm), with $\gamma = 2\beta\cdot\widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n^{1.5})$.

Proof. First, suppose that adversary $\mathcal{A}$ can break the computational binding property of the commitment scheme COM with non-negligible probability. 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 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 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 $\|\mathbf{x}\|_\infty \le 2\beta$ and $\mathbf{C}\cdot\mathbf{x} = \mathbf{0} \mod q$.

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 $\mathbf{u} = \mathbf{C}\cdot\mathbf{z} \mod q$.
2. Run $\mathsf{TrapGen}(n,m,q)$ algorithm ℓ 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 $\mathbf{A} = \big[\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1\big]\in\mathbb{Z}_q^{n\times(2\ell+1)\cdot m}$, 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 $\mathbf{x}_i^{1-d^*[i]} = \mathbf{0}^m$,
   - $\mathsf{grt}[d^*] = \mathbf{A}_0\cdot\mathbf{x}_0 \mod q\in\mathbb{Z}_q^n$.
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 $d[b]$ $(1\le b\le\ell)$ 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 ℓ 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
   $$\mathbf{t}^{(d)} = \mathbf{u} - \big(\mathbf{A}_0\cdot\mathbf{x}_0 + \sum_{i\in[\ell],i\neq b}\mathbf{A}_i^{d[i]}\cdot\mathbf{x}_i^{d[i]}\big) \mod q.$$
   - 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 let $\mathbf{x}^{(d)} = \big(\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\ldots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1\big)\in\mathbb{Z}^{(2\ell+1)\cdot m}$. 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 $\mathsf{grt}[d] = \mathbf{A}_0\cdot\mathbf{x}_0 \mod q\in\mathbb{Z}_q^n$.
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 $\varSigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$. Queries to the random oracle $\mathcal{H}$ are handled by consistently returning uniformly random values in $\{1,2,3\}^t$. For each $\kappa \leq q_{\mathcal{H}}$, we let r_κ denote the answer to the κ-th query.

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

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

such that $\mathsf{Verify}(\mathsf{gpk}, RL^*, \varSigma^*, M^*) = \mathsf{valid}$, and the implicit tracing algorithm fails or traces to a user outside of the coalition $U \setminus RL^*$. Now algorithm $\mathcal{F}$ exploits the forgery as follows.

First, one can argue that $\mathcal{A}$ must have queried $\mathcal{H}$ on input $\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$, as otherwise, the probability that $(Ch_1, \ldots, Ch_t) = \mathcal{H}\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$ is at most 3^{-t}. Therefore, with probability at least $\epsilon - 3^{-t}$, there exists certain $\kappa^* \leq q_{\mathcal{H}}$ such that the κ^*-th oracle queries involves the tuple $\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$. Next, $\mathcal{F}$ picks κ^* 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 κ^*-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 [35, Lemma 7] implies that, with probability larger than 1/2, algorithm $\mathcal{F}$ can obtain a 3-fork involving the tuple $\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$ 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 CMT_i. Since COM is assumed to be computationally-binding, we can apply Lemma 2 to extract a vector $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ satisfying $\mathbf{A} \cdot \mathbf{y} = \mathbf{u} \bmod q$, $\mathbf{A}_0 \cdot \mathbf{y}_0 \bmod q \notin RL^*$, and $\mathbf{y} \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$. Now consider two cases:

- If $d \neq d^*$, which happens with probability at most $\frac{N-1}{N}$, then algorithm $\mathcal{F}$ declares 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 $\varSigma^*$ is a valid signature such that the implicit tracing algorithm either fails or outputs an index $e \notin U \setminus RL^*$.

- If the tracing algorithm fails, then, in particular, one has $\mathsf{Verify}(\mathsf{gpk}, \mathsf{grt}[d^*], \varSigma^*, M^*) = \mathsf{Valid}$. It follows from the correctness of the VLR group signature that $\mathbf{A}_0 \cdot \mathbf{y}_0 \neq \mathsf{grt}[d^*] = \mathbf{A}_0 \cdot \mathbf{z}_0$. This implies that $\mathbf{y}_0 \neq \mathbf{z}_0$, and thus $\mathbf{y}^* \neq \mathbf{z}$.

- If the tracing algorithm outputs $e \notin U \setminus RL^*$, namely the following two facts simultaneously hold true:

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

  This leads to $\mathsf{grt}[e] \notin RL^*$, and hence $e \notin U$. Furthermore, the correctness of the revocation check and the computational binding property of COM imply that $\mathbf{A}_0 \cdot \mathbf{y}_0 \bmod q = \mathsf{grt}[e]$. Now consider 2 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 1), 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 $d^* \neq e$ (because $e \notin U$) , and thus $\mathsf{grt}[d^*] \neq \mathsf{grt}[e]$. In other words, we have $\mathbf{A}_0 \cdot \mathbf{y}_0 \neq \mathbf{A}_0 \cdot \mathbf{z}_0 \bmod q$. This leads to $\mathbf{y}^* \neq \mathbf{z}$.

Now let $\mathbf{x} = \mathbf{z} - \mathbf{y}^* \in \mathbb{Z}^{(\ell+1)m}$, then $\mathbf{x} \neq \mathbf{0}$; $\mathbf{C} \cdot \mathbf{x} = \mathbf{0} \bmod q$; and $\|\mathbf{x}\|_\infty \leq \|\mathbf{z}\|_\infty + \|\mathbf{y}\|_\infty \leq \beta + \beta = 2\beta$. 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. The authors would like to thank D. Stehlé, B. Libert, R. Bhattacharyya, J. Chen, and the anonymous reviewers for their helpful comments. The research is supported in part by the Singapore Ministry of Education under Research Grant MOE2013-T2-1-041. Adeline Langlois is supported in part by ERC Starting Grant ERC-2013-StG-335086-LATTAC.

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*, volume 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. 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.
18. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, pages 197–206. ACM, 2008.
19. 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.
20. J. Groth. Fully Anonymous Group Signatures Without Random Oracles. In *ASIACRYPT*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
21. 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.
22. 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.
23. 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.
24. 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.
25. 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.
26. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
27. V. Lyubashevsky. Lattice Signatures without Trapdoors. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.
28. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
29. D. Micciancio and O. Regev. Lattice-based Cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.
30. 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.
31. 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.
32. 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.
33. C. Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In *CRYPTO*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010.
34. 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.
35. 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.
36. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC*, pages 84–93. ACM, 2005.
37. M. Rückert. Adaptively Secure Identity-Based Identification from Lattices without Random Oracles. In *SCN*, volume 6280 of *LNCS*, pages 345–362. Springer, 2010.
38. 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.
39. J. Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

A Witness Extraction

The following lemma says that in our protocol, one can extract a satisfying witness under specific conditions.

Lemma 3. *Assuming that for a given commitment* CMT*, there exist* 3 valid *responses* $\mathrm{RSP}^{(1)}$*,* $\mathrm{RSP}^{(2)}$*, and* $\mathrm{RSP}^{(3)}$ *corresponding to all* 3 *possible values of the challenge Ch. If* COM *is a computationally binding commitment scheme, then one can efficiently extract a vector* $\mathbf{y} = \big(\mathbf{y}_0\|\mathbf{y}_1^0\|\mathbf{y}_1^1\|\ldots\|\mathbf{y}_\ell^0\|\mathbf{y}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ *satisfying* $\mathbf{A}\cdot\mathbf{y} = \mathbf{u} \bmod q$*,* $\mathbf{y} \in \mathsf{Secret}_\beta(d)$ *for some* $d \in \{0,1\}^\ell$*, and* $\mathbf{A}_0 \cdot \mathbf{y}_0 \bmod q \notin RL$*.*

Proof. Let $\mathrm{CMT} = \big(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\big) \in (\mathbb{Z}_q^n)^4$, and let $\mathrm{RSP}^{(1)}$, $\mathrm{RSP}^{(2)}$, $\mathrm{RSP}^{(3)}$ as in (3), (4), and (5), respectively. Since all 3 responses satisfy the verification conditions, the followings are true:

$$\begin{cases} \forall j \in [p]:\ \mathbf{v}_j \in \mathsf{SecretExt}(d_1);\ \ \mathbf{c}_0 = \mathsf{COM}(d_3,\ \{\psi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{j,0}\big) \bmod q); \\ \forall \mathbf{u}_i \in RL:\ \mathbf{c}_0 \neq \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big) - \mathbf{u}_i \bmod q\big); \\ \mathbf{c}_1 = \mathrm{COM}\Big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j\big) - \mathbf{u}\Big) = \mathrm{COM}\Big(d_3,\ \{\psi_j\}_{j=1}^p,\ \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_j\big)\Big); \\ \mathbf{c}_2 = \mathrm{COM}\big(\{\mathbf{w}_j\}_{j=1}^p\big) = \mathrm{COM}\Big(\{\mathrm{T}_{d_3} \circ \psi_j(\mathbf{h}_j)\}_{j=1}^p\big)\Big); \\ \mathbf{c}_3 = \mathrm{COM}\big(\{\mathbf{v}_j + \mathbf{w}_j\}_{j=1}^p\big) = \mathrm{COM}\Big(\{\mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j)\}_{j=1}^p\big)\Big). \end{cases}$$

Since COM is computationally binding, one can deduce that $d_2 = d_3$, $\phi_j = \psi_j$ for all $j \in [p]$, and that:

$$\begin{cases} \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_{j,0} - \mathbf{h}_{j,0})\big) \notin RL, \\ \forall j \in [p]:\ \mathbf{w}_j = \mathrm{T}_{d_2} \circ \phi_j(\mathbf{h}_j) \text{ and } \ \mathbf{v}_j + \mathbf{w}_j = \mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j), \\ \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_j - \mathbf{h}_j)\big) = \mathbf{u} \bmod q. \end{cases}$$

For each $j \in [p]$, let $\mathbf{y}'_j = \mathbf{s}_j - \mathbf{h}_j$, then $\mathrm{T}_{d_2} \circ \phi_j(\mathbf{y}'_j) = \mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j) - \mathrm{T}_{d_2} \circ \phi_j(\mathbf{h}_j) = \mathbf{v}_j \in \mathsf{SecretExt}(d_1)$. It then follows that $\phi_j(\mathbf{y}'_j) \in \mathsf{SecretExt}(d_1 \oplus d_2)$. Let $d = d_1 \oplus d_2$, then $\mathbf{y}'_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, since the permutation $\phi_j \in \mathcal{S}$ 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 $\mathbf{y} \in \mathsf{Secret}_\beta(d)$.

Let $\mathbf{y} = \big(\mathbf{y}_0\|\mathbf{y}_1^0\|\mathbf{y}_1^1\|\ldots\|\mathbf{y}_\ell^0\|\mathbf{y}_\ell^1\big)$, then the blocks $\mathbf{y}_1^{1-d[1]}, \ldots, \mathbf{y}_\ell^{1-d[\ell]}$ are zero-blocks $\mathbf{0}^m$. Furthermore, we have that:

$$\mathbf{A}_0 \cdot \mathbf{y}_0 = \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_{j,0} - \mathbf{h}_{j,0})\big) \notin RL.$$

Finally, by construction, we have: $\mathbf{A} \cdot \mathbf{y} = \mathbf{A}^* \cdot \mathbf{y}' = \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{y}_j = \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot (\mathbf{s}_j - \mathbf{h}_j)\big) = \mathbf{u} \bmod q$. Therefore, we have obtained a vector $\mathbf{y}$ satisfying all the conditions stated in the lemma. □

B Analysis of our scheme

B.1 Correctness

Theorem 6. *Our VLR group signature scheme is correct with overwhelming probability.*

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

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

1. We first prove that: $\mathsf{grt}[d] \notin RL \Rightarrow \mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid}$.

Suppose that $\mathsf{grt}[d] \notin RL$. We will show that, for each $k \in [t]$, all the checks performed by the verification algorithm hold true, except for negligible probability. For simplicity, we will not consider the trivial checks for correct computations, e.g., the case $Ch^{(k)} = 3$.

(a) If $Ch^{(k)} = 1$: The crucial point is to check whether $\forall j \in [p] : \mathbf{v}_j^{(k)} \in \mathsf{SecretExt}(d_1^{(k)})$. Note that if $\mathbf{x} = \mathsf{gsk}[d]$ is outputted by $\mathsf{KeyGen}(n, N)$ then $\mathbf{x} \in \mathsf{Secret}_\beta(d)$, and thus, all the vectors $\mathbf{z}_1, \ldots, \mathbf{z}_p$ outputted by the procedure $\mathsf{WitnessDE}(\mathbf{x})$ belong to the set $\mathsf{SecretExt}(d)$. It then follows from the special properties of the permutation sets $\mathcal{S}$ and $\mathcal{T}$ that $\forall j \in [p] : \mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j) \in \mathsf{SecretExt}(d \oplus e^{(k)})$. Finally, it is worth to recall that $\forall j \in [p] : \mathbf{v}_j^{(k)} = \mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j)$, and that $d_1^{(k)} = d \oplus e^{(k)}$.

(b) If $Ch^{(k)} = 2$: There are two crucial checks:

i. Check if $\forall \mathbf{u}_i \in RL : \mathbf{c}_0 \neq \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big) - \mathbf{u}_i \bmod q\big)$. For each i, let $\alpha_i = \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big) - \mathbf{u}_i \in \mathbb{Z}_q^n$. Meanwhile, $\mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \alpha)$, where $\alpha = \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big) = \alpha_i + \mathbf{u}_i - \mathsf{grt}[d]$. Since $\mathsf{grt}[d] \notin RL$, we have $\mathsf{grt}[d] \neq \mathbf{u}_i$ for all i, and thus, $\alpha \neq \alpha_i$. Moreover, over the randomness of all algorithms, the distributions of $\mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \alpha)$ and $\mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \alpha_i)$ are statistically close to uniform over $\mathbb{Z}_q^n$ (this follows from the statistically hiding property of COM). Hence, we have $\mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \alpha) \neq \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \alpha_i)$ with overwhelming probability.

ii. Check if $\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j^{(k)}\big) - \mathbf{u} = \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)}\big)$. This is true, because

$$\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j^{(k)}\big) = \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot \big(\mathbf{z}_j + \mathbf{r}_j^{(k)}\big) = \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{z}_j + \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)} = \mathbf{u} + \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)},$$

where the last equation follows from the fact that $\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j\big) = \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.

Therefore, the verification algorithm outputs Valid with overwhelming probability, over the randomness of all algorithms.

2. We then prove that: $\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid} \Rightarrow \mathsf{grt}[d] \notin RL$.

Assume by contradiction that $\mathsf{grt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q \in RL$, and fix any $k \in [t]$. Note that in the signing algorithm, we construct $\mathbf{c}_0^{(k)}$ so that:

$$\mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big) \bmod q\big)$$

On the other hand, since the verification algorithm outputs Valid, the following requirement must satisfy (in the case $Ch^{(k)} = 2$):

$$\mathbf{c}_0^{(k)} \neq \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ ,\mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big) - \mathbf{u}_i \bmod q\big)$$

As we have $\mathbf{s}_{j,0}^{(k)} = \mathbf{z}_{j,0} + \mathbf{r}_{j,0}^{(k)}$ and $\mathbf{A}_0 \cdot \mathbf{x}_0 = \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_{j,0}\big)$, we have that $\mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(k)}\big) = \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}^{(k)}\big) - \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q$. Thus, we obtain a contradiction. Namely, it must be true that $\mathsf{grt}[d] \notin RL$. This concludes the proof. □

B.2 Selfless-anonymity

Lemma 4. *The signature Σ^* outputted by Game G_1 is a valid signature, and is statistically indistinguishable from the legitimate signature Σ produced by Game G_0.*

Proof. Let

$$\Sigma^* = \big(M^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$$

be the signature outputted by Game G_1. First of all, we observe that:

- For every $k \in [t]$, the distribution of $\mathrm{CMT}^{(k)}$ is statistically close to uniform over $(\mathbb{Z}_q^n)^4$. This follows from the statistical regularity property of $f_{\mathbf{B}}$ and the statistically hiding property of COM.
- The distribution of $\big(Ch^{(1)}, \ldots, Ch^{(t)}\big)$ is uniform over $\{1, 2, 3\}^t$.

Therefore, the distributions of $\{\mathrm{CMT}^{(k)}\}_{k=1}^t$ and $\{Ch^{(k)}\}_{k=1}^t$ are statistically close to those of the legitimate signature Σ. We now will show that for every $k \in [t]$, $\mathrm{RSP}^{(k)}$ is statistically close to that of the legitimate signature, and it is valid 'response' to $\mathrm{CMT}^{(k)}$ and $Ch^{(k)}$. Indeed, for each $k \in [t]$, we have:

1. If $Ch^{(k)} = 1$, then the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big)$ and $\mathrm{RSP}^{(k)}$ is one of the following two cases:
   (a)

$$\begin{cases} \mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)\big), \\ \mathbf{c}_1^{(k)} = \mathrm{COM}\big(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^p,\ \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)}\big)\big), \\ \mathbf{c}_2^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\}_{j=1}^p\big), \\ \mathbf{c}_3^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)})\}_{j=1}^p\big), \end{cases} \tag{13}$$

   and

$$\mathrm{RSP}^{(k)} = \big(d^{(k)} \oplus e^{(k)},\ \big\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})\big\}_{j=1}^p,\ \big\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\big\}_{j=1}^p\big). \tag{14}$$

   For all $j \in [p]$, since $\mathbf{z}_j^{(k)} \in \mathsf{SecretExt}(d^{(k)})$, it follows from (1) that $\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)}) \in \mathsf{SecretExt}(d^{(k)} \oplus e^{(k)})$. Thus $\mathrm{RSP}^{(k)}$ satisfies the verification conditions for the case $Ch^{(k)} = 1$ (since the checks with respect to $\mathbf{c}_2^{(k)}$ and $\mathbf{c}_3^{(k)}$ obviously hold true). Note that by construction, $d^{(k)} \oplus e^{(k)}$ is uniform in $\{0,1\}^\ell$; $\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})$ is uniform in $\mathsf{SecretExt}(d^{(k)} \oplus e^{(k)})$; and $\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})$ is uniform in $\mathbb{Z}_q^{(2\ell+1)3m}$. Therefore, the distribution of $\mathrm{RSP}^{(k)}$ is identical to that of the legitimate signature.

   (b)

$$\begin{cases} \mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \mathbf{A}_0 \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)\big), \\ \mathbf{c}_1^{(k)} = \mathrm{COM}_{\mathbf{B}}\big(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^p,\ \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot (\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}) - \mathbf{u}\big), \\ \mathbf{c}_2^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\}_{j=1}^p\big), \\ \mathbf{c}_3^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)})\}_{j=1}^p\big), \end{cases} \tag{15}$$

   and $\mathrm{RSP}^{(k)}$ is computed as in (14). The analysis for this case is similar to the above one.

2. If $Ch^{(k)} = 2$, then the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big)$ and $\mathrm{RSP}^{(k)}$ is one of the following two cases:
   (a) $\mathrm{CMT}^{(k)}$ is computed as in (13), and $\mathrm{RSP}^{(k)} = \big(e^{(k)}, \ \{\pi_j^{(k)}\}_{j=1}^p, \ \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^p\big)$. Observe that:
      - By construction, we have $\mathbf{A}_0 \cdot \mathbf{g}_0 \notin RL$. The correctness of the VLR group signature then implies that: the revocation check with respect to $\mathbf{c}_0^{(k)}$ holds true with overwhelming probability.
      - By construction, we have $\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j^{(k)}\big) = \mathbf{u} \bmod q$. This implies that the check with respect to $\mathbf{c}_1^{(k)}$ holds true.
      - The check with respect to $\mathbf{c}_3^{(k)}$ obviously hold true.

      Hence $\mathrm{RSP}^{(k)}$ satisfies the verification conditions for the case $Ch^{(k)} = 2$. Moreover, $\mathrm{RSP}^{(k)}$ is uniform over $\{0,1\}^\ell \times \mathcal{S}^p \times \big(\mathbb{Z}_q^{(2\ell+1)3m}\big)^p$, and thus, is identically distributed with that of the legitimate signature.

   (b) $\mathrm{CMT}^{(k)}$ is computed as in (15), and $\mathrm{RSP}^{(k)} = \big(e^{(k)}, \ \{\pi_j^{(k)}\}_{j=1}^p, \ \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^p\big)$. As above, the distribution of $\mathrm{RSP}^{(k)}$ is the same as in the legitimate signature. Moreover:
      - Since we have $\mathbf{A}_0 \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{z}_{j,0}^{(k)}) \notin RL$, the revocation check with respect to $\mathbf{c}_0^{(k)}$ holds true with overwhelming probability.
      - We remark that we do not have $\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j^{(k)}\big) = \mathbf{u} \bmod q$, but we construct $\mathbf{c}_1^{(k)}$ so that the check with respect to it holds true.
      - The check with respect to $\mathbf{c}_3^{(k)}$ obviously hold true.
3. If $Ch^{(k)} = 3$, then in any of the two views of the adversary, the verification checks with respect to $\mathbf{c}_1^{(k)}$, and $\mathbf{c}_2^{(k)}$ are checks for correct computations, and thus, they hold true. Moreover, the distribution of $\mathrm{RSP}^{(k)}$ is uniform over $\{0,1\}^\ell \times \mathcal{S}^p \times \big(\mathbb{Z}_q^{(2\ell+1)3m}\big)^p$, as in the legitimate signature.

Hence, we have shown that the simulated signature Σ^* produced by game G_1 is a valid signature of M^* under gpk and RL, and it is statistically close to the legitimate signature Σ produced by game G_0. □