Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions

Benoît Libert[1], San Ling[2], Fabrice Mouhartem[1], Khoa Nguyen[2], and Huaxiong Wang[2]

[1] Ecole Normale Supérieure de Lyon (France)
[2] Nanyang Technological University (Singapore)

Abstract. A recent line of works – initiated by Gordon, Katz and Vaikuntanathan (Asiacrypt 2010) – gave lattice-based constructions allowing users to authenticate while remaining hidden in a crowd. Despite five years of efforts, known constructions are still limited to static sets of users, which cannot be dynamically updated. This work provides new tools enabling the design of anonymous authentication systems whereby new users can join the system at any time.

Our first contribution is a signature scheme with efficient protocols, which allows users to obtain a signature on a committed value and subsequently prove knowledge of a signature on a committed message. This construction is well-suited to the design of anonymous credentials and group signatures. It indeed provides the first lattice-based group signature supporting dynamically growing populations of users.

As a critical component of our group signature, we provide a simple joining mechanism of introducing new group members using our signature scheme. This technique is combined with zero-knowledge arguments allowing registered group members to prove knowledge of a secret short vector of which the corresponding public syndrome was certified by the group manager. These tools provide similar advantages to those of structure-preserving signatures in the realm of bilinear groups. Namely, they allow group members to generate their own public key without having to prove knowledge of the underlying secret key. This results in a two-message joining protocol supporting concurrent enrollments, which can be used in other settings such as group encryption.

Our zero-knowledge arguments are presented in a unified framework where: (i) The involved statements reduce to arguing possession of a $\{-1,0,1\}$-vector $\mathbf{x}$ with a particular structure and satisfying $\mathbf{P}\cdot\mathbf{x}=\mathbf{v} \bmod q$ for some public matrix $\mathbf{P}$ and vector $\mathbf{v}$; (ii) The reduced statements can be handled using permuting techniques for Stern-like protocols. Our framework can serve as a blueprint for proving many other relations in lattice-based cryptography.

Keywords. Lattice-based cryptography, anonymity, signatures with efficient protocols, dynamic group signatures, anonymous credentials.

1 Introduction

Lattice-based cryptography is currently emerging as a promising alternative to traditional public-key techniques. During the last decade, it has received a permanent interest due to its numerous advantages. Not only does it seemingly resist quantum attacks, it also provides a better asymptotic efficiency than its relatives based on conventional number theory. While enabling advanced functionalities (like fully homomorphic [42] or attribute-based/predicate encryption [45,46] for circuits) that remain elusive outside the lattice world, lattice-based primitives tend to interact with zero-knowledge proofs [44] less smoothly than their counterparts in abelian groups endowed with a bilinear map (see, e.g., [18,32,39,50,2]) or groups of hidden order [6,30,31,27]. Arguably, this partially arises from the fact that lattices have far less algebraic structure than, e.g., pairing-friendly cyclic groups. It is not surprising that the most efficient zero-knowledge proofs for lattice-related languages [15] take advantage of the extra algebraic structure available in the ring setting [64]. A consequence of the scarcity of truly efficient zero-knowledge proofs in the lattice setting is that, in the context of anonymity and privacy-preserving protocols, lattice-based cryptography has undergone significantly slower development than in other areas like functional encryption [45,46]. While natural realizations of ring signatures [70] showed up promptly [53,22] after the seminal work of Gentry, Peikert and Vaikuntanathan (GPV) [43], viable constructions of lattice-based group signatures remained lacking until the work of Gordon, Katz and Vaikuntanathan [47] in 2010. Despite recent advances in the area [58,14,66,62], lattice-based group signatures and other privacy-preserving primitives remain substantially less practical and powerful in terms of functionalities than their siblings based on traditional number theoretic problems [6,18,39,56] for which solutions even exist [20,21,49,10] outside the random oracle methodology. For example, we still have no convenient realization of group signature supporting dynamic groups [13,56] or anonymous credentials [35,29].

In this paper, we address the latter two problems by first proposing a lattice-based signature with efficient protocols in the fashion of Camenisch and Lysyanskaya [31]. To ease its use in the design of dynamic group signatures, we introduce a zero-knowledge argument system that allows a user to prove knowledge of a signature on a public key for which the user knows the underlying secret key.

Related Work. Anonymous credentials were first suggested by Chaum [35] and efficiently realized by Camenisch and Lysyanskaya [29,31]. They involve one or more credential issuer(s) and a set of users who have a long-term secret key which constitutes a their digital identity and pseudonyms that can be seen as commitments to their secret key. Users can dynamically obtain credentials from an issuer that only knows users' pseudonyms and obliviously certifies users' secret keys as well as (optionally) a set of attributes. Later on, users can make themselves known to verifiers under a different pseudonym and demonstrate possession of the issuer's signature on their secret key without revealing neither the signature or the key. Anonymous credentials typically consist of a protocol whereby the user obtains the issuer's signature on a committed message, another protocol for proving that two commitments open to the same value (which allows proving that the same secret underlies two distinct pseudonyms) and a protocol for proving possession of a secret message-signature pair.

The first efficient constructions were given by Camenisch and Lysyanskaya under the Strong RSA assumption [29,31] or using bilinear groups [32]. Other solutions were subsequently given with additional useful properties such as non-interactivity [10], delegatability [9] or support for efficient attributes [25] (see [28] and references therein). Anonymous credentials with attributes are often obtained by having the issuer obliviously sign a multi-block message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, where one block is the secret key while other blocks contain public or private attributes. Note that, for the sake of keeping the scheme compatible with zero-knowledge proofs, the blocks $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ cannot be simply hashed before getting signed using a ordinary, single-block signature.

Group signatures are a central anonymity primitive, introduced by Chaum and van Heyst [36] in 1991, which allows members of a group managed by some authority to sign messages in the name of the entire group. At the same time, users remain accountable for the messages they sign since an opening authority can identify them if they misbehave.

Ateniese, Camenisch, Joye and Tsudik [6] provided the first scalable construction meeting the security requirements that can be intuitively expected from the primitive, although clean security notions were not available yet at that time. Bellare, Micciancio and Warinschi [11] filled this gap by providing suitable security notions for static groups, which were subsequently extended to the dynamic setting[3] by Kiayias and Yung [56] and Bellare, Shi and Zhang [13]. In these models, efficient schemes have been put forth in the random oracle model [56,39] (the ROM) and in the standard model [49,2,1].

Lattice-based group signatures were put forth for the first time by Gordon, Katz and Vaikuntanathan [47] whose solution had linear-size signatures in the number of group members. Camenisch, Neven and Rückert [33] extended [47] so as to achieve anonymity in the strongest sense. Laguillaumie *et al.* [57] decreased the signature length to be logarithmic in the number N_{gs} of group members. While asymptotically shorter, their signatures remained space-consuming as, analogously to the Boyen-Waters group signature [20], their scheme encrypts each bit of the signer's identity individually. Simpler and more efficient solutions with $\mathcal{O}(\log N)$ signature size were given by Nguyen, Zhang and Zhang [66] and Ling, Nguyen and Wang [62]. In particular, the latter scheme [62] achieves significantly smaller signatures by encrypting all bits of the signer's identity at once. Benhamouda *et al.* [14] described a hybrid group signature that simultaneously relies on lattice assumptions (in the ring setting) and discrete-logarithm-related assumptions. Recently, Libert, Ling, Nguyen and Wang [60] obtained substantial efficiency improvements via a construction based on Merkle trees which eliminates

[3] By "dynamic setting", we refer to a scenario where new group members can register at any time but, analogously to [13,56], we do not consider the orthogonal problem of user revocation here.

the need for GPV trapdoors [43]. For the time being, all known group signatures are designed for static groups and analyzed in the model of Bellare, Micciancio and Warinschi [11], where no new group member can be introduced after the setup phase. To date, it remains an open problem to design a lattice-based system that supports dynamically growing population of users in the models of [13,56].

Our Contributions. Our first result is a lattice-based signature with efficient protocols for multi-block messages. Namely, we provide a way for a user to obtain a signature on a committed N-block message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ as well as a protocol for proving possession of a valid message-signature pair. The signature and its companion protocols can serve as a building block for lattice-based anonymous credentials and can potentially find applications in other privacy-preserving protocols (e.g., [26]) based on lattice assumptions. The main application that we consider in this paper is the design of a lattice-based group signature scheme for dynamic groups. While less efficient than the recent proposal of [60], our scheme is competitive with the solution of Ling, Nguyen and Wang [62] which is one of the most efficient candidates based on standard (i.e., non-ideal) lattices in static groups. In particular, it features $\mathcal{O}(\log N_{\mathsf{gs}})$-size signatures for groups of up to N_{gs} members and only lengthens the signatures of [62] by a (small) constant factor. We prove the security of our system in the random oracle model [12] under the Short Integer Solution (SIS) and Learning With Errors (LWE) assumptions.

As a stepping stone in the design of our dynamic group signature, we also develop a zero-knowledge argument system allowing a group member to prove knowledge of a secret key (made of a short Gaussian vector) and a membership certificate issued by the group manager on the corresponding public key. Analogously to structure-preserving signatures [2], our signature scheme and zero-knowledge arguments make it possible to sign public keys without hashing them while remaining oblivious of the underlying secret key. They thus enable a round-optimal dynamic joining protocol – which allows the group manager to introduce new group members by issuing a membership certificate on their public key – which does not require any proof of knowledge on behalf of the prospective user. As a result, the interaction is minimal: only one message is sent in each direction between the prospective user and the group manager.[4] Besides being the first lattice-based group signature for dynamic groups, our scheme thus remains secure in the setting advocated by Kiayias and Yung [55], where many users want to join the system at the same time and concurrently interact with the group manager. We believe that, analogously to structure-preserving signatures [2,1], the combination of our signature scheme and zero-knowledge arguments can serve as a building blocks for other primitives, including group encryption [54] or adaptive oblivious transfer [48].

Our Techniques. Our signature scheme with efficient protocols builds on the SIS-based signature of Böhl *et al.* [16], which is itself a variant of Boyen's

[4] Note that each signature still requires the user to prove knowledge of his secret key. However, this is not a problem in concurrent settings as the argument of knowledge is made non-interactive via the Fiat-Shamir heuristic.

signature [19]. Recall that the latter scheme involves a public key containing matrices $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n \times m}$ and signs an ℓ-bit message $\mathfrak{m} \in \{0,1\}^\ell$ by computing a short $\mathbf{v} \in \mathbb{Z}^{2m}$ such that $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \mathfrak{m}[i]\mathbf{A}_j] \cdot \mathbf{v} = \mathbf{0}^n \bmod q$. The variant proposed by Böhl *et al.* [16] only uses a constant number of matrices $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n \times m}$. Each signature is associated with a single-use tag tag (which is only used in one signing query in the proof) and the public key involves an extra matrix $\mathbf{D} \in \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. A message Msg is signed by first applying a chameleon hash function $\mathbf{h} = \mathsf{CMHash}(\mathsf{Msg}, \mathbf{s}) \in \{0,1\}^m$ and signing $\mathbf{h}$ by computing a short $\mathbf{v} \in \mathbb{Z}^m$ such that $[\mathbf{A} \mid \mathbf{A}_0 + \mathsf{tag} \cdot \mathbf{A}_1] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathbf{h} \bmod q$.

Our scheme extends [16] – modulo the use of a larger number of matrices $(\{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \{\mathbf{D}\}_{k=0}^{N})$ – so that an N-block message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N) \in (\{0,1\}^L)^N$, for some $L \in \mathbb{N}$, is signed by outputting a tag $\tau \in \{0,1\}^\ell$ and a short $\mathbf{v} \in \mathbb{Z}^{2m}$ such that $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{CMHash}(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s})$, where the chameleon hash function computes $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \bmod q$, for some short vector $\mathbf{s}$, before re-encoding $\mathbf{c}_M$ so as to enable multiplication by $\mathbf{D}$.

In order to obtain a signature scheme akin to the one of Camenisch and Lysyanksaya [31], our idea is to have the tag $\tau \in \{0,1\}^\ell$ play the same role as the prime exponent in Strong-RSA-based schemes [31]. In the security proof of [16], we are faced with two situations: either the adversary produces a signature on a fresh tag $\tau^\star$, or it recycles a tag $\tau^{(i)}$ used by the signing oracle for a new, un-signed message $(\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$. In the former case, the proof can proceed as in Boyen's proof [19]. In the latter case, the reduction must guess upfront which tag $\tau^{(i^\dagger)}$ the adversary will choose to re-use and find a way to properly answer the $i^\dagger$-th signing query without using the vanished trapdoor (for other queries, the Agrawal *et al.* technique [3] applies to compute a suitable $\mathbf{v}$ using a trapdoor hidden in $\{\mathbf{A}_j\}_{j=0}^{\ell}$). Böhl *et al.* [16] solve this problem by "programming" the vector $\mathbf{u} \in \mathbb{Z}_q^n$ in a special way and achieve full security using chameleon hashing.

To adapt this idea in the context of signatures with efficient protocols, we have to overcome several difficulties. The first one is to map $\mathbf{c}_M$ back in the domain of the chameleon hash function while preserving the compatibility with zero-knowledge proofs. To solve this problem, we extend a technique used in [60] in order to build a "zero-knowledge-friendly" chameleon hash function. This function hashes $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ by outputting the coordinate-wise binary decomposition $\mathbf{w}$ of $\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k$. If we define the "powers-of-2" matrix $\mathbf{H} = \mathbf{I} \otimes \left[1 \mid 2 \mid \ldots \mid 2^{\lceil \log q \rceil}\right]$, then we can prove that $\mathbf{w} = \mathsf{CMHash}(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s})$ by demonstrating the knowledge of short vectors $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s}, \mathbf{w})$ such that $\mathbf{H} \cdot \mathbf{w} = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \bmod q$, which boils down to arguing knowledge of a solution to the ISIS problem [61].

The second problem is to prove knowledge of $(\tau, \mathbf{v}, \mathbf{s})$ and $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ satisfying $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{CMHash}(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s})$, without revealing any of the witnesses. To this end, we provide a framework for proving all the involved statement (and many other relations that naturally arise in lattice-based cryptography) as special cases. We reduce the statements to asserting that a short integer vector $\mathbf{x}$ satisfies an equation of the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$, for some public matrix $\mathbf{P}$ and vector $\mathbf{v}$, and belongs to a set VALID of short vectors with a particular structure. While the small-norm property of $\mathbf{x}$ is provable using standard techniques (e.g., [63]), we argue its membership of VALID by leveraging the properties of Stern-like protocols [72,53,61]. In particular, we rely on the fact that their underlying permutations interact well with combinatorial statements pertaining to $\mathbf{x}$, especially $\mathbf{x}$ being a bitstring with a specific pattern. We believe our framework to be of independent interest as it provides a blueprint for proving many other intricate relations in a modular manner.

When we extend the scheme with a protocol for signing committed messages, we need the signer to re-randomize the user's commitment before signing the hidden messages. This is indeed necessary to provide the reduction with a backdoor allowing to correctly answer the $i^{\dagger}$-th query by "programming" the randomness of the commitment. Since we work with integers vectors, a straightforward simulation incurs a non-negligible statistical distance between the simulated distributions of re-randomization coins and the real one (which both have a discrete Gaussian distribution). Camenisch and Lysyanskaya [31] address a similar problem by choosing the signer's randomness to be exponentially larger than that of the user's commitment so as to statistically "drown" the aforementioned discrepancy. Here, the same idea would require to work with an exponentially large modulus q. Instead, we adopt a more efficient solution, inspired by Bai *et al.* [7], which is to apply an analysis based on the Rényi divergence rather than the statistical distance. In short, the Rényi divergence's properties tell us that, if some event E occurs with noticeable probability in some probability space P, so does it in a different probability space Q for which the second order divergence $R_2(P||Q)$ is sufficiently small. In our setting, $R_2(P||Q)$ is precisely polynomially bounded since the two probability spaces only diverge in one signing query.

Our dynamic group signature scheme avoids these difficulties because the group manager only signs known messages: instead of signing the user's secret key as in anonymous credentials, it creates a membership certificate by signing the user's public key. Our zero-knowledge arguments accommodate the requirements of the scheme in the following way. In the joining protocol that dynamically introduces new group members, the user i chooses a membership secret consisting of a short discrete Gaussian vector $\mathbf{z}_i$. This user generates a public syndrome $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \mod q$, for some public matrix $\mathbf{F}$, which constitutes his public key. In order to certify $\mathbf{v}_i$, the group manager computes the coordinate-wise binary expansion $\mathsf{bin}(\mathbf{v}_i)$ of $\mathbf{v}_i$. The vector $\mathsf{bin}(\mathbf{v}_i)$ is then signed using our signature scheme. Using the resulting signature $(\tau, \mathbf{v}, \mathbf{s})$ as a membership certificate, the group member is able to sign a message by proving that: (i) He holds a valid signature $(\tau, \mathbf{v}, \mathbf{s})$ on some secret binary message $\mathsf{bin}(\mathbf{v}_i)$; (ii) The latter vector $\mathsf{bin}(\mathbf{v}_i)$ is the binary expansion of some syndrome $\mathbf{v}_i$ of which he knows a GPV pre-image $\mathbf{z}_i$. We remark that condition (ii) can be proved by providing evidence that we have $\mathbf{v}_i = \mathbf{H} \cdot \mathsf{bin}(\mathbf{v}_i) = \mathbf{F} \cdot \mathbf{z}_i \bmod q$, for some short integer vector $\mathbf{z}_i$ and some binary $\mathsf{bin}(\mathbf{v}_i)$, where $\mathbf{H}$ is the "powers-of-2" matrix. Our abstraction of Stern-like protocols [72,53,61] allows us to efficiently argue such statements. The fact that the underlying chameleon hash function smoothly interacts with

Stern-like zero-knowledge arguments is the property that maintains the user's capability of efficiently proving knowledge of the underlying secret key.

Organization. In the forthcoming sections, we first provide some background in Section 2. Our signature with efficient protocols is presented in Section 3, where we also give protocols for obtaining a signature on a committed message and proving possession of a message-signature pair. Section 4 uses our signature scheme in the design of a dynamic group signature. The details of the zero-knowledge arguments used in Section 3 and Section 4 are deferred to Section 5, where we present them in a unified framework.

2 Background and Definitions

In the following, all vectors are denoted in bold lower-case letters, whereas bold upper-case letters will be used for matrices. If $\mathbf{b} \in \mathbb{R}^n$, its Euclidean norm and infinity norm will be denoted by $\|\mathbf{b}\|$ and $\|\mathbf{b}\|_\infty$, respectively. The Euclidean norm of matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ with columns $(\mathbf{b}_i)_{i \leq n}$ is denoted by $\|\mathbf{B}\| = \max_{i \leq n} \|\mathbf{b}_i\|$. If $\mathbf{B}$ is full column-rank, we let $\widetilde{\mathbf{B}}$ denote its Gram-Schmidt marginalization.

When S is a finite set, we denote by $U(S)$ the uniform distribution over S and by $x \hookleftarrow U(S)$ the action of sampling x according to this distribution.

2.1 Lattices

A (full-rank) lattice L is the set of all integer linear combinations of some linearly independent basis vectors $(\mathbf{b}_i)_{i \leq n}$ belonging to some $\mathbb{R}^n$. We work with q-ary lattices, for some prime q.

Definition 1. *Let $m \geq n \geq 1$, a prime $q \geq 2$ and $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, define $\Lambda_q(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \mid \exists \mathbf{s} \in \mathbb{Z}_q^n \ \ s.t. \ \ \mathbf{A}^T \cdot \mathbf{s} = \mathbf{e} \bmod q\}$ as well as*

$$\Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{0}^n \bmod q\}, \quad \Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}$$

For any $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$, $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{t}$ so that $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a shift of $\Lambda_q^{\perp}(\mathbf{A})$.

For a lattice L, a vector $\mathbf{c} \in \mathbb{R}^n$ and a real $\sigma > 0$, define $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/\sigma^2)$. The discrete Gaussian distribution of support L, parameter σ and center $\mathbf{c}$ is defined as $D_{L,\sigma,\mathbf{c}}(\mathbf{y}) = \rho_{\sigma,\mathbf{c}}(\mathbf{y})/\rho_{\sigma,\mathbf{c}}(L)$ for any $\mathbf{y} \in L$. We denote by $D_{L,\sigma}(\mathbf{y})$ the distribution centered in $\mathbf{c} = \mathbf{0}$. We will extensively use the fact that samples from $D_{L,\sigma}$ are short with overwhelming probability.

Lemma 1 ([8, Le. 1.5]). *For any lattice $L \subseteq \mathbb{R}^n$ and positive real number $\sigma > 0$, we have $\Pr_{\mathbf{b} \hookleftarrow D_{L,\sigma}}[\|\mathbf{b}\| \leq \sqrt{n}\sigma] \geq 1 - 2^{-\Omega(n)}$.*

As shown by Gentry *et al.* [43], Gaussian distributions with lattice support can be sampled from efficiently, given a sufficiently short basis of the lattice.

Lemma 2 ([23, Le. 2.3]). *There exists a* PPT *(probabilistic polynomial-time) algorithm* $\mathsf{GPVSample}$ *that takes as inputs a basis $\mathbf{B}$ of a lattice $L \subseteq \mathbb{Z}^n$ and a rational $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$, and outputs vectors $\mathbf{b} \in L$ with distribution $D_{L,\sigma}$.*

Lemma 3 ([4, Th. 3.2]). *There exists a* PPT *algorithm* $\mathsf{TrapGen}$ *that takes as inputs* 1^n, 1^m *and an integer* $q \geq 2$ *with* $m \geq \Omega(n \log q)$, *and outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{n \times m})$, *and* $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.

Lemma 3 is often combined with the sampler from Lemma 2. Micciancio and Peikert [65] recently proposed a more efficient approach for this combined task, which should be preferred in practice but, for the sake of simplicity, we present our schemes using $\mathsf{TrapGen}$.

We also make use of an algorithm that extends a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ to a trapdoor of any $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ whose left $n \times m$ submatrix is $\mathbf{A}$.

Lemma 4 ([34, Le. 3.2]). *There exists a* PPT *algorithm* $\mathsf{ExtBasis}$ *that takes as inputs a matrix* $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ *whose first* m *columns span* $\mathbb{Z}_q^n$, *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *where* $\mathbf{A}$ *is the left* $n \times m$ *submatrix of* $\mathbf{B}$, *and outputs a basis* $\mathbf{T_B}$ *of* $\Lambda_q^{\perp}(\mathbf{B})$ *with* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$.

In our security proofs, analogously to [19,16] we also use a technique due to Agrawal, Boneh and Boyen [3] that implements an all-but-one trapdoor mechanism (akin to the one of Boneh and Boyen [17]) in the lattice setting.

Lemma 5 ([3, Th. 19]). *There exist a* PPT *algorithm* $\mathsf{SampleRight}$ *that takes as input matrices* $\mathbf{A}, \mathbf{C} \in \mathbb{Z}_q^{n \times m}$, *a low-norm matrix* $\mathbf{R} \in \mathbb{Z}^{m \times m}$, *a short basis* $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^{\perp}(\mathbf{C})$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$ *and a rational* σ *such that* $\sigma \geq \|\widetilde{\mathbf{T_C}}\| \cdot \Omega(\sqrt{\log n})$, *and outputs vectors* $\mathbf{b} \in \mathbb{Z}^{2m}$ *such that* $\big[\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C}\big] \cdot \mathbf{b} = \mathbf{u} \bmod q$ *and with distribution statistically close to* $D_{L,\sigma}$ *where* L *denotes the shifted lattice* $\{\mathbf{x} \in \mathbb{Z}^{2m} : \big[\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C}\big] \cdot \mathbf{x} = \mathbf{u} \bmod q\}$.

2.2 Computational Problems

The security of our schemes provably relies (in the ROM) on the assumption that both algorithmic problems below are hard, i.e., cannot be solved in polynomial time with non-negligible probability and non-negligible advantage, respectively.

Definition 2. *Let* m, q, β *be functions of a parameter* n. *The Short Integer Solution problem* $\mathsf{SIS}_{n,m,q,\beta}$ *is as follows: Given* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, *find* $\mathbf{x} \in \Lambda_q^{\perp}(\mathbf{A})$ *with* $0 < \|\mathbf{x}\| \leq \beta$.

If $q \geq \sqrt{n}\beta$ and $m, \beta \leq \mathsf{poly}(n)$, then $\mathsf{SIS}_{n,m,q,\beta}$ is at least as hard as standard worst-case lattice problem SIVP_γ with $\gamma = \widetilde{\mathcal{O}}(\beta\sqrt{n})$ (see, e.g., [43, Se. 9]).

Definition 3. *Let* $n, m \geq 1$, $q \geq 2$, *and let* χ *be a probability distribution on* $\mathbb{Z}$. *For* $\mathbf{s} \in \mathbb{Z}_q^n$, *let* $A_{\mathbf{s},\chi}$ *be the distribution obtained by sampling* $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ *and* $e \hookleftarrow \chi$, *and outputting* $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. *The Learning With Errors problem* $\mathsf{LWE}_{n,q,\chi}$ *asks to distinguish* m *samples chosen according to* $\mathcal{A}_{\mathbf{s},\chi}$ *(for* $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$*) and* m *samples chosen according to* $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

If q is a prime power, $B \geq \sqrt{n}\omega(\log n)$, $\gamma = \widetilde{\mathcal{O}}(nq/B)$, then there exists an efficient sampleable B-bounded distribution χ (i.e., χ outputs samples with norm at most B with overwhelming probability) such that $\mathsf{LWE}_{n,q,\chi}$ is as least as hard as SIVP_γ (see, e.g., [69,68,23]).

3 A Lattice-Based Signature with Efficient Protocols

Our scheme can be seen as a variant of the Böhl *et al.* signature [16], where each signature is a triple $(\tau, \mathbf{v}, \mathbf{s})$, made of a tag $\tau \in \{0,1\}^\ell$ and integer vectors $(\mathbf{v}, \mathbf{s})$ satisfying $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathbf{h} \bmod q$, where $\mathbf{A}, \mathbf{D} \in \mathbb{Z}_q^{n \times m}$ are public random matrices and $\mathbf{h} \in \{0,1\}^m$ is a chameleon hash of the message which is computed using randomness $\mathbf{s}$. A difference is that, while [16] uses a short single-use tag $\tau \in \mathbb{Z}_q$, we need the tag to be an ℓ-bit string $\tau \in \{0,1\}^\ell$ which will assume the same role as the prime exponent of Camenisch-Lysyanskaya signatures [31] in the security proof.

We show that a suitable chameleon hash function makes the scheme compatible with Stern-like zero-knowledge arguments [61,62] for arguing possession of a valid message-signature pair. Section 5 shows how to translate such a statement into asserting that a short witness vector $\mathbf{x}$ with a particular structure satisfies a relation of the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$, for some public matrix $\mathbf{P}$ and vector $\mathbf{v}$. The underlying chameleon hash can be seen as a composition of the chameleon hash of [34, Section 4.1] with a technique used in [67,60]: on input of a message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, it outputs the binary decomposition of $\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k$, for some discrete Gaussian vector $\mathbf{s}$.

3.1 Description

We assume that messages are vectors of N blocks $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, where each block is a $2m$-bit string $\mathfrak{m}_k = \mathfrak{m}_k[1] \ldots \mathfrak{m}_k[2m] \in \{0,1\}^{2m}$ for $k \in \{1, \ldots, N\}$.

For each vector $\mathbf{v} \in \mathbb{Z}_q^L$, we denote by $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{L\lceil \log q \rceil}$ the vector obtained by replacing each coordinate of $\mathbf{v}$ by its binary representation.

Keygen$(1^\lambda, 1^N)$**:** Given a security parameter $\lambda > 0$ and the number of blocks $N = \mathsf{poly}(\lambda)$, choose the following parameters: $n = \mathcal{O}(\lambda)$; a prime modulus $q = \widetilde{\mathcal{O}}(N \cdot n^4)$; dimension $m = 2n\lceil \log q \rceil$; an integer $\ell = \Theta(\lambda)$; and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q} \log n)$, $\sigma_0 = 2\sqrt{2}(N+1)\sigma m^{3/2}$, and $\sigma_1 = \sqrt{\sigma_0^2 + \sigma^2}$. Define the message space as $(\{0,1\}^{2m})^N$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$. This basis allows computing short vectors in $\Lambda_q^\perp(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{n \times m})$.
2. Choose random matrices $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_N \hookleftarrow U(\mathbb{Z}_q^{2n \times 2m})$ as well as a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

The private key consists of $SK := \mathbf{T}_\mathbf{A}$ and the public key is

$$PK := \big(\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \{\mathbf{D}_k\}_{k=0}^{N},\ \mathbf{D},\ \mathbf{u}\big).$$

$\mathsf{Sign}\big(SK, \mathsf{Msg}\big)$: To sign an N-block message $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N) \in \big(\{0,1\}^{2m}\big)^N$,

1. Choose a random binary string $\tau \hookleftarrow U(\{0,1\}^\ell)$. Then, using $SK := \mathbf{T}_{\mathbf{A}}$, compute a short delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the matrix
$$\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}. \qquad (1)$$
2. Choose a discrete Gaussian vector $\mathbf{s} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_1}$. Compute the vector $\mathbf{c}_M \in \mathbb{Z}_q^{2n}$ as a chameleon hash of $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$. Namely, compute
$$\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^{2n},$$
which is used to define $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M) \in \mathbb{Z}_q^n$. Then, using the delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, sample a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_\tau),\sigma}$.

Output the signature $sig = (\tau, \mathbf{v}, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$.

$\mathsf{Verify}\big(PK, \mathsf{Msg}, sig\big)$: Given PK, a message $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N) \in (\{0,1\}^{2m})^N$ and a purported signature $sig = (\tau, \mathbf{v}, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$, return 1 if
$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k) \bmod q. \qquad (2)$$
and $\|\mathbf{v}\| < \sigma\sqrt{2m}$, $\|\mathbf{s}\| < \sigma_1\sqrt{2m}$.

When the scheme is used for obliviously signing committed messages, the security proof follows Bai *et al.* [7] in that it applies an argument based on the Rényi divergence in one signing query. This argument requires to sample $\mathbf{s}$ from a Gaussian distribution whose standard deviation σ_1 is polynomially larger than σ.

We note that, instead of being included in the public key, the matrices $\{\mathbf{D}_k\}_{k=0}^N$ can be part of common public parameters shared by many signers. Indeed, only the matrices $(\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell})$ should be specific to the user who holds the secret key $SK = \mathbf{T}_{\mathbf{A}}$. In Section 3.3, we use a variant where $\{\mathbf{D}_k\}_{k=0}^N$ belong to public parameters.

3.2 Security Analysis

The security analysis in Theorem 1 requires that $q > \ell$.

Theorem 1. *The scheme is secure under chosen-message attacks under the* SIS *assumption.*

Proof. The prove the result, we will distinguish two kinds of attacks:

Type I attacks are attacks where, in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, $\tau^\star$ did not appear in any output of the signing oracle.

Type II attacks are such that, in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, $\tau^\star$ is recycled from an output $sig^{(i^\star)} = (\tau^{(i^\star)}, \mathbf{v}^{(i^\star)}, \mathbf{s}^{(i^\star)})$ of the signing oracle, for some index $i^\star \in \{1, \ldots, Q\}$. However, if $\mathsf{Msg}^\star = (\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$ and $\mathsf{Msg}^{(i^\star)} = (\mathfrak{m}_1^{(i^\star)}, \ldots, \mathfrak{m}_N^{(i^\star)})$ denote the forgery message and the $i^\star$-th signing query, respectively, we have $\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^\star \neq \mathbf{D}_0 \cdot \mathbf{s}^{(i^\star)} + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\star)}$.

Type III attacks are those where the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$ recycles $\tau^\star$ from an output $sig^{(i^\star)} = (\tau^{(i^\star)}, \mathbf{v}^{(i^\star)}, \mathbf{s}^{(i^\star)})$ of the signing oracle (i.e., $\tau^{(i^\star)} = \tau^\star$ for some index $i^\star \in \{1, \ldots, Q\}$) and we have the collision

$$\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^\star = \mathbf{D}_0 \cdot \mathbf{s}^{(i^\star)} + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\star)}. \tag{3}$$

Type III attacks imply a collision for the chameleon hash function of Kawachi *et al.* [53]: if (3) holds, a short vector of $\Lambda_q^\perp([\mathbf{D}_0 \mid \mathbf{D}_1 \mid \ldots \mid \mathbf{D}_N])$ is obtained as

$$\big(\mathbf{s}^{\star T} - \mathbf{s}^{(i^\star)^T} \mid \mathfrak{m}_1^{\star T} - \mathfrak{m}_1^{(i^\star)^T} \mid \ldots \mid \mathfrak{m}_N^{\star}{}^T - \mathfrak{m}_N^{(i^\star)^T}\big)^T,$$

so that a collision breaks the SIS assumption.

The security against Type I attacks is proved by Lemma 6 which applies the same technique as in [19,65]. In particular, the prefix guessing technique of [51] allows keeping the modulus smaller than the number Q of adversarial queries as in [65]. In order to deal with Type II attacks, we can leverage the technique of [16]. In Lemma 7, we prove that Type II attack would also contradict SIS. □

Lemma 6. *The scheme is secure against Type I attacks if the* $\mathsf{SIS}_{n,m,q,\beta'}$ *assumption holds for* $\beta' = m^{3/2}\sigma^2(\ell+3) + m^{1/2}\sigma_1$. (The proof is given in Appendix B.2.)

Lemma 7. *The scheme is secure against Type II attacks if the* $\mathsf{SIS}_{n,m,q,\beta''}$ *assumption holds for* $\beta'' = \sqrt{2}(\ell+2)\sigma^2 m^{3/2} + m^{1/2}$. (The proof is detailed in Appendix B.3.)

3.3 Protocols for Signing a Committed Value and Proving Possession of a Signature

We first show a two-party protocol whereby a user can interact with the signer in order to obtain a signature on a committed message.

In order to prove that the scheme still guarantees unforgeability for obliviously signed messages, we will assume that each message block $\mathfrak{m}_k \in \{0,1\}^{2m}$ is obtained by encoding the actual message $M_k = M_k[1] \ldots M_k[m] \in \{0,1\}^m$ as $\mathfrak{m}_k = \mathsf{Encode}(M_k) = (\bar{M}_k[1], M_k[1], \ldots, \bar{M}_k[m], M_k[m])$. Namely, each 0 (resp. each 1) is encoded as a pair $(1,0)$ (resp. $(0,1)$). The reason for this encoding is that the proof of Theorem 2 requires that at least one block $\mathfrak{m}_k^\star$ of the forgery message is 1 while the same bit is 0 at some specific signing query. We will show (see Section 5) that the correctness of this encoding can be efficiently proved using Stern-like [72] protocols.

To sign committed messages, a first idea is exploit the fact that our signature of Section 3.1 blends well with the SIS-based commitment scheme suggested by Kawachi *et al.* [53]. In the latter scheme, the commitment key consists of matrices $(\mathbf{D}_0, \mathbf{D}_1) \in \mathbb{Z}_q^{2n \times 2m} \times \mathbb{Z}_q^{2n \times 2m}$, so that message $\mathfrak{m} \in \{0,1\}^{2m}$ can be committed to by sampling a Gaussian vector $\mathbf{s} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and computing $\mathbf{C} = \mathbf{D}_0 \cdot \mathbf{s} + \mathbf{D}_1 \cdot \mathfrak{m} \in \mathbb{Z}_q^{2n}$. This scheme extends to commit to multiple messages $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ at once by computing $\mathbf{C} = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^{2n}$ using a longer commitment key $(\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_N) \in (\mathbb{Z}_q^{2n \times 2m})^{N+1}$. It is easy to see that the resulting commitment remains statistically hiding and computationally binding under the SIS assumption.

In order to make our construction usable in the definitional framework of Camenisch *et al.* [28], we assume common public parameters (i.e., a common reference string) and encrypt all witnesses of which knowledge is being proved under a public key included in the common reference string. The resulting ciphertexts thus serve as statistically binding commitments to the witnesses. To enable this, the common public parameters comprise public keys $\mathbf{G}_0 \in \mathbb{Z}_q^{n \times \ell}$, $\mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m}$ for multi-bits variants of the dual Regev cryptosystem [43] and all parties are withheld access to the underlying private keys. The flexibility of Stern-like protocols allows us to prove that the content of a perfectly hiding commitment $\mathbf{c}_{\mathfrak{m}}$ is consistent with encrypted values.

Global-Setup: Let $B = \sqrt{n}\omega(\log n)$ and let χ be a B-bounded distribution. Let $p = \sigma \cdot \omega(\sqrt{m})$ upper-bound entries of vectors sampled from the distribution $D_{\mathbb{Z}^{2m},\sigma}$. Generate two public keys for the dual Regev encryption scheme in its multi-bit variant. These keys consists of a public random matrix $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ and random matrices $\mathbf{G}_0 = \mathbf{B} \cdot \mathbf{E}_0 \in \mathbb{Z}_q^{n \times \ell}$, $\mathbf{G}_1 = \mathbf{B} \cdot \mathbf{E}_1 \in \mathbb{Z}_q^{n \times 2m}$, where $\mathbf{E}_0 \in \mathbb{Z}^{m \times \ell}$ and $\mathbf{E}_1 \in \mathbb{Z}^{m \times 2m}$ are short Gaussian matrices with columns sampled from $D_{\mathbb{Z}^m,\sigma}$. These matrices will be used to encrypt integer vectors of dimension ℓ and $2m$, respectively. Finally, generate public parameters $CK := \{\mathbf{D}_k\}_{k=0}^{N}$ consisting of uniformly random matrices $\mathbf{D}_k \hookleftarrow U(\mathbb{Z}_q^{2n \times 2m})$ for a statistically hiding commitment to vectors in $(\{0,1\}^{2m})^N$. Return public parameters consisting of

$$\mathsf{par} := \{\ \mathbf{B} \in \mathbb{Z}_q^{n \times m},\ \mathbf{G}_0 \in \mathbb{Z}_q^{n \times \ell},\ \mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m},\ CK\}.$$

Protocol: The signer S, who holds a key pair $PK := \{\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{D},\ \mathbf{u}\}$, $SK := \mathbf{T_A}$, interacts with the user U who has a message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, in the following interactive protocol.

1. U samples $\mathbf{s}' \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and computes $\mathbf{c}_{\mathfrak{m}} = \mathbf{D}_0 \cdot \mathbf{s}' + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^{2n}$ which is sent to S as a commitment to $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$. In addition, U encrypts $\{\mathfrak{m}_k\}_{k=1}^{N}$ and $\mathbf{s}'$ under the dual-Regev public key $(\mathbf{B}, \mathbf{G}_1)$ by computing for all $k \in \{1, \ldots, N\}$:

$$\begin{aligned}\mathbf{c}_k &= (\mathbf{c}_{k,1}, \mathbf{c}_{k,2}) \\ &= \big(\mathbf{B}^T \cdot \mathbf{s}_{k,0} + \mathbf{e}_{k,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_{k,0} + \mathbf{e}_{k,2} + \mathfrak{m}_k \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \qquad (4)\end{aligned}$$

for randomly chosen $\mathbf{s}_{k,0} \hookleftarrow \chi^n$, $\mathbf{e}_{k,1} \hookleftarrow \chi^m$, $\mathbf{e}_{k,2} \hookleftarrow \chi^{2m}$, and

$$\begin{aligned}\mathbf{c}_{s'} &= (\mathbf{c}_{s',1}, \mathbf{c}_{s',2}) \\ &= \big(\mathbf{B}^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,2} + \mathbf{s}' \cdot \lfloor q/p \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \qquad (5)\end{aligned}$$

where $\mathbf{s}_0 \hookleftarrow \chi^n$, $\mathbf{e}_{0,1} \hookleftarrow \chi^m$, $\mathbf{e}_{0,2} \hookleftarrow \chi^{2m}$. The ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$ and $\mathbf{c}_{s'}$ are sent to S along with $\mathbf{c}_{\mathfrak{m}}$.

Then, U generates an interactive zero-knowledge argument to convince S that $\mathbf{c}_{\mathfrak{m}}$ is a commitment to $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ with the randomness $\mathbf{s}'$ such that $\{\mathfrak{m}_k\}_{k=1}^N$ and $\mathbf{s}'$ were honestly encrypted to $\{\mathbf{c}_k\}_{i=1}^N$ and $\mathbf{c}_{s'}$, as in (4) and (5). For convenience, this argument system will be described in Section 5.3, where we demonstrate that, together with other zero-knowledge protocols used in this work, it can be derived from a Stern-like [72] protocol constructed in Section 5.1.

2. If the argument of step 1 properly verifies, S samples $\mathbf{s}'' \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_0}$ and computes a vector $\mathbf{u}_{\mathfrak{m}} = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{c}_{\mathfrak{m}} + \mathbf{D}_0 \cdot \mathbf{s}''\big) \in \mathbb{Z}_q^n$. Next, S randomly picks $\tau \hookleftarrow \{0,1\}^\ell$ and uses $\mathbf{T}_{\mathbf{A}}$ to compute a delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the matrix $\mathbf{A}_\tau \in \mathbb{Z}_q^{n \times 2m}$ of (1). Using $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, S samples a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_\tau),\sigma}$. It returns the vector $(\tau, \mathbf{v}, \mathbf{s}'') \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$ to U.
3. U computes $\mathbf{s} = \mathbf{s}' + \mathbf{s}''$ over $\mathbb{Z}$ and verifies that

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k\big) \bmod q.$$

If so, it outputs $(\tau, \mathbf{v}, \mathbf{s})$. Otherwise, it outputs $\perp$.

Note that, if both parties faithfully run the protocol, the user obtains a valid signature $(\tau, \mathbf{v}, \mathbf{s})$ for which the distribution of $\mathbf{s}$ is $D_{\mathbb{Z}^{2m},\sigma_1}$, where $\sigma_1 = \sqrt{\sigma^2 + \sigma_0^2}$.

In order to prove possession of a message-signature pair, the user runs the following protocol.

Prove: On input of a signature $(\tau, \mathbf{v} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$ on the message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, the user does the following.

1. Using $(\mathbf{B}, \mathbf{G}_0)$ and $(\mathbf{B}, \mathbf{G}_1)$ generate perfectly binding commitments to $\tau \in \{0,1\}^\ell$, $\{\mathfrak{m}_k\}_{k=1}^N$, $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}^m$ and $\mathbf{s} \in \mathbb{Z}^{2m}$. Namely, compute

$$\begin{aligned}\mathbf{c}_\tau &= (\mathbf{c}_{\tau,1}, \mathbf{c}_{\tau,2}) \\ &= \big(\mathbf{B}^T \cdot \mathbf{s}_\tau + \mathbf{e}_{\tau,1},\ \mathbf{G}_0^T \cdot \mathbf{s}_\tau + \mathbf{e}_{\tau,2} + \tau \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell, \\ \mathbf{c}_k &= (\mathbf{c}_{k,1}, \mathbf{c}_{k,2}) \\ &= \big(\mathbf{B}^T \cdot \mathbf{s}_{k,0} + \mathbf{e}_{k,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_{k,0} + \mathbf{e}_{k,2} + \mathfrak{m}_k \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall k \in \{1, \ldots, N\}\end{aligned}$$

for where $\mathbf{s}_\tau \hookleftarrow \chi^n$, $\mathbf{e}_{\tau,1} \hookleftarrow \chi^m$, $\mathbf{e}_{\tau,2} \hookleftarrow \chi^\ell$, $\mathbf{s}_{k,0} \hookleftarrow \chi^n$, $\mathbf{e}_{k,1} \hookleftarrow \chi^m$, $\mathbf{e}_{k,2} \hookleftarrow \chi^{2m}$, as well as

$$\begin{aligned}
\mathbf{c}_{\mathbf{v}_1} &= (\mathbf{c}_{\mathbf{v}_1,1}, \mathbf{c}_{\mathbf{v}_1,2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_{\mathbf{v}_1} + \mathbf{e}_{\mathbf{v}_1,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_{\mathbf{v}_1} + \mathbf{e}_{\mathbf{v}_1,2} + \mathbf{v}_1 \cdot \lfloor q/p \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^m \\
\mathbf{c}_{\mathbf{v}_2} &= (\mathbf{c}_{\mathbf{v}_2,1}, \mathbf{c}_{\mathbf{v}_2,2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_{\mathbf{v}_2} + \mathbf{e}_{\mathbf{v}_2,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_{\mathbf{v}_2} + \mathbf{e}_{\mathbf{v}_2,2} + \mathbf{v}_2 \cdot \lfloor q/p \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^m \\
\mathbf{c}_s &= (\mathbf{c}_{s,1}, \mathbf{c}_{s,2}) \\
&= \big(\mathbf{B}^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,2} + \mathbf{s} \cdot \lfloor q/p \rfloor\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m},
\end{aligned}$$

where $\mathbf{s}_{\mathbf{v}_1}, \mathbf{s}_{\mathbf{v}_2} \hookleftarrow \chi^n$, $\mathbf{e}_{\mathbf{v}_1,1}, \mathbf{e}_{\mathbf{v}_1,2} \hookleftarrow \chi^m$, $\mathbf{e}_{\mathbf{v}_2,1}, \mathbf{e}_{\mathbf{v}_2,2} \hookleftarrow \chi^m$, $\mathbf{s}_0 \hookleftarrow \chi^n$, $\mathbf{e}_{0,1} \hookleftarrow \chi^m$, $\mathbf{e}_{0,2} \hookleftarrow \chi^{2m}$.

2. Prove in zero-knowledge that $\mathbf{c}_\tau$, $\mathbf{c}_s$, $\mathbf{c}_{\mathbf{v}_1}$, $\mathbf{c}_{\mathbf{v}_2}$, $\{\mathbf{c}_k\}_{k=1}^N$ encrypt a valid message-signature pair. In Section 5.4, we show that this involved zero-knowledge protocol can be derived from the statistical zero-knowledge argument of knowledge for a simpler, but more general relation that we explicitly present in Section 5.1. The proof system can be made statistically ZK for a malicious verifier using standard techniques (assuming a common reference string, we can use [37]). In the random oracle model, it can be made non-interactive using the Fiat-Shamir heuristic [41].

We require that the adversary be unable to prove possession of a signature of a message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ for which it did not legally obtain a credential by interacting with the issuer. Note that the messages that are blindly signed by the issuer are uniquely defined since, at each signing query, the adversary is required to supply perfectly binding commitments $\{\mathbf{c}_k\}_{k=1}^N$ to $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$.

In instantiations using interactive proofs, we do not consider it as a forgery if the adversary simply replays a proof generated by a honest prover.

The proof of Theorem 2 makes crucial use of the Rényi divergence using arguments in the spirit of Bai *et al.* [7]. The reduction has to guess upfront the index $i^\star \in \{1, \ldots, Q\}$ of the specific signing query for which the adversary will re-use $\tau^{(i^\star)}$. For this query, the reduction will have to make sure that the simulation trapdoor of Agrawal *et al.* [3] (used by the $\mathsf{SampleRight}$ algorithm of Lemma 5) vanishes: otherwise, the adversary's forgery would not be usable for solving SIS. This means that, as in the proof of [16], the reduction must answer exactly one signing query in a different way, without using the trapdoor. While Böhl *et al.* solve this problem by exploiting the fact that they only need to prove security against non-adaptive forgers, we directly use a built-in chameleon hash function mechanism which is implicitly realized by the matrix $\mathbf{D}_0$ and the vector $\mathbf{s}$. Namely, in the signing query for which the Agrawal *et al.* trapdoor [3] cancels, we assign a special value to the vector $\mathbf{s} \in \mathbb{Z}^{2m}$, which depends on the adaptively-chosen signed message $(\mathsf{Msg}_1^{(i^\star)}, \ldots, \mathsf{Msg}_N^{(i^\star)})$ and some Gaussian matrices $\{\mathbf{R}_k\}_{k=1}^N$ hidden behind $\{\mathbf{D}_k\}_{k=1}^N$.

One issue is that this results in a different distribution for the vector $\mathbf{s} \in \mathbb{Z}^m$. However, we can still view $\mathbf{s}$ as a vector sampled from a Gaussian distribution centered away from $\mathbf{0}^{2m}$. Since this specific situation occurs only once during the simulation, we can apply a result proved in [59] which upper-bounds the Rényi divergence between two Gaussian distributions with identical standard deviations but different centers. By choosing the standard deviation σ_1 of $\mathbf{s} \in \mathbb{Z}^{2m}$ to be polynomially larger than that of the columns of matrices $\{\mathbf{R}_k\}_{k=1}^{N}$, we can keep the Rényi divergence between the two distributions of $\mathbf{s}$ (i.e., the one of the simulation and the one of the real game) sufficiently small to apply the probability preservation property (which still gives a polynomial reduction since the argument must only be applied on one signing query). Namely, the latter implies that, if the Rényi divergence $R_2(\mathbf{s}^{\mathsf{real}}||\mathbf{s}^{\mathsf{sim}})$ is polynomial, the probability that the simulated vector $\mathbf{s}^{\mathsf{sim}} \in \mathbb{Z}^{2m}$ passes the verification test will only be polynomially smaller than in the real game and so will be the adversary's probability of success.

Another option would have been to keep the statistical distance between $\mathbf{s}^{\mathsf{real}}$ and $\mathbf{s}^{\mathsf{sim}}$ negligible using the smudging technique of [5]. However, this would have implied to use an exponentially large modulus q since σ_1 should have been exponentially larger than the standard deviations of the columns of $\{\mathbf{R}_k\}_{k=1}^{N}$.

Theorem 2. *Under the* $\mathsf{SIS}_{n,2m,q,\hat{\beta}}$ *assumption, where* $\hat{\beta} = N\sigma(2m)^{3/2}+4\sigma_1 m^{3/2}$*, the above protocols are secure protocols for obtaining a signature on a committed message and proving possession of a valid message-signature pair.* (The proof is given in Appendix B.4.)

Theorem 3. *The scheme provides anonymity under the* $\mathsf{LWE}_{n,q,\chi}$ *assumption.* (The proof is given in Appendix B.5.)

4 A Dynamic Lattice-Based Group Signature

In this section, the signature scheme of Section 3 is used to design a group signature for dynamic groups using the syntax and the security model of Kiayias and Yung [56], which is recalled in Appendix A.

In the notations hereunder, for any positive integers $\mathfrak{n}$, and $q \geq 2$, we define the "powers-of-2" matrix $\mathbf{H}_{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil} \in \mathbb{Z}_q^{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil}$ as:

$$\begin{aligned}\mathbf{H}_{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil} &= \mathbf{I}_{\mathfrak{n}} \otimes [1 \mid 2 \mid 4 \mid \ldots \mid 2^{\lceil\log q\rceil-1}] \\ &= \begin{bmatrix} 1\ 2\ 4\ \ldots\ 2^{\lceil\log q\rceil-1} & & & \\ & 1\ 2\ 4\ \ldots\ 2^{\lceil\log q\rceil-1} & & \\ & & \ddots & \\ & & & 1\ 2\ 4\ \ldots\ 2^{\lceil\log q\rceil-1} \end{bmatrix}.\end{aligned}$$

Also, for each vector $\mathbf{v} \in \mathbb{Z}_q^{\mathfrak{n}}$, we define $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{\mathfrak{n}\lceil\log q\rceil}$ to be the vector obtained by replacing each entry of $\mathbf{v}$ by its binary expansion. Hence, we have $\mathbf{v} = \mathbf{H}_{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil} \cdot \mathsf{bin}(\mathbf{v})$ for any $\mathbf{v} \in \mathbb{Z}_q^{\mathfrak{n}}$.

In our scheme, each group membership certificate is a signature generated by the group manager on the user's public key. Since the group manager only needs to sign known (rather than committed) messages, we can use a simplified version of the signature, where the chameleon hash function does not need to choose the discrete Gaussian vector $\mathbf{s}$ with a larger standard deviation than other vectors.

A key component of the scheme is the two-message joining protocol whereby the group manager admits new group members by signing their public key. The first message is sent by the new user $\mathcal{U}_i$ who samples a membership secret consisting of a short vector $\mathbf{z}_i \hookleftarrow D_{\mathbb{Z}^{4m},\sigma}$ (where $m = 2n\lceil\log q\rceil$), which is used to compute a syndrome $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$ for some public matrix $\mathbf{F} \in \mathbb{Z}_q^{4n \times 4m}$. This syndrome $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$ must be signed by $\mathcal{U}_i$ using his long term secret key $\mathsf{usk}[i]$ (as in [56,13], we assume that each user has a long-term key $\mathsf{upk}[i]$ for a digital signature, which is registered in some PKI) and will uniquely identify $\mathcal{U}_i$. In order to generate a membership certificate for $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$, the group manager GM signs its binary expansion $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{4n\lceil\log q\rceil}$ using the scheme of Section 3.

Equipped with his membership certificate $(\tau, \mathbf{d}, \mathbf{s}) \in \{0,1\}^{\ell} \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$, the new group member $\mathcal{U}_i$ can sign a message using a Stern-like protocol for demonstrating his knowledge a valid certificate for which he also knows the secret key associated with the certified public key $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$. This boils down to providing evidence that the membership certificate is a valid signature on some binary message $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{4n\lceil\log q\rceil}$ for which he also knows a short $\mathbf{z}_i \in \mathbb{Z}^{4m}$ such that $\mathbf{v}_i = \mathbf{H}_{4n\times 2m} \cdot \mathsf{bin}(\mathbf{v}_i) = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$.

Interestingly, the process does not require any proof of knowledge of the membership secret $\mathbf{z}_i$ during the joining phase, which is round-optimal. Analogously to the Kiayias-Yung technique [55] and constructions based on structure-preserving signatures [2], the joining protocol thus remains secure in environments where many users want to register at the same time in concurrent sessions.

We remark that a similar Stern-like protocol could also be directly used to prove knowledge of a Boyen signature [19] on a binary expansion of the user's syndrome $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$ while preserving the user's ability to prove knowledge of a short $\mathbf{z}_i \in \mathbb{Z}^{4m}$ such that $\mathbf{F} \cdot \mathbf{z}_i = \mathbf{v}_i \bmod q$. However, this would require considerably longer private keys containing $4n \cdot \log q$ matrices $\{\mathbf{A}_j\}_{j=0}^{\ell}$ of dimension $n \times m$ each (i.e., we would need $\ell = \Theta(n \cdot \log q)$). In contrast, by using the signature scheme of Section 3, we only need the group public key $\mathcal{Y}$ to contain $\ell = \log N_{\mathsf{gs}}$ matrices in $\mathbb{Z}_q^{n\times m}$. Since the number of users N_{gs} is polynomial, we have $\log N_{\mathsf{gs}} \ll n$, which results in a much more efficient scheme.

4.1 Description of the Scheme

$\mathsf{Setup}(1^{\lambda}, 1^{N_{\mathsf{gs}}})$: Given a security parameter $\lambda > 0$ and the maximal expected number of group members $N_{\mathsf{gs}} = 2^{\ell} \in \mathsf{poly}(\lambda)$, choose lattice parameter $n = \mathcal{O}(\lambda)$; prime modulus $q = \widetilde{\mathcal{O}}(\ell n^3)$; dimension $m = 2n\lceil\log q\rceil$; Gaussian parameter $\sigma = \Omega(\sqrt{n\log q}\log n)$; infinity norm bounds $\beta = \sigma\omega(\log m)$ and $B = \sqrt{n}\omega(\log n)$. Let χ be a B-bounded distribution. Choose a hash function $H : \{0,1\}^* \to \{1,2,3\}^t$ for some $t = \omega(\log n)$, which will be modeled as a random oracle in the security analysis. Then, do the following.

1. Generate a key pair for the signature scheme of Section 3.1 for signing single-block messages. Namely, run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$

and a short basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$. This basis allows computing short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ with Gaussian parameter σ. Next, choose random matrices $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$, $\mathbf{D}_0, \mathbf{D}_1 \hookleftarrow U(\mathbb{Z}_q^{2n\times 2m})$ and a vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

2. Choose an additional random matrix $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{4n\times 4m})$ uniformly. Looking ahead, this matrix will be used to ensure security against framing attacks.
3. Generate a master key pair for the Gentry-Peikert-Vaikuntanathan IBE scheme in its multi-bit variant. This key pair consists of a statistically uniform matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T_B} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(\mathbf{B})$. This basis will allow us to compute GPV private keys with a Gaussian parameter $\sigma_{\mathrm{GPV}} \geq \|\widetilde{\mathbf{T}}_{\mathbf{B}}\| \cdot \sqrt{\log m}$.
4. Choose a one-time signature scheme $\Pi^{\mathrm{OTS}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ and a hash function $H_0 : \{0,1\}^* \to \mathbb{Z}_q^{n\times 2m}$, that will be modeled as random oracles in the security analysis.

The group public key is defined as

$$\mathcal{Y} := \big(\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{B},\ ,\ \mathbf{D},\ \mathbf{D}_0,\ \mathbf{D}_1,\ \mathbf{F},\ \mathbf{u},\ \Pi^{\mathrm{OTS}},\ H,\ H_0\big).$$

The opening authority's private key is $\mathcal{S}_{\mathsf{OA}} := \mathbf{T_B}$ and the private key of the group manager consists of $\mathcal{S}_{\mathsf{GM}} := \mathbf{T_A}$. The algorithm outputs $\big(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}\big)$.

Join$^{(\mathsf{GM},\mathcal{U}_i)}$: the group manager GM and the prospective user $\mathcal{U}_i$ run the following interactive protocol: $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$

1. $\mathcal{U}_i$ samples a discrete Gaussian vector $\mathbf{z}_i \leftarrow D_{\mathbb{Z}^{4m},\sigma}$ and computes $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$. He sends the vector $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$, whose binary representation $\mathsf{bin}(\mathbf{v}_i)$ consists of $4n\lceil \log q \rceil = 2m$ bits, together with an ordinary digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}(\mathbf{v}_i)$ to GM.
2. J_{GM} verifies that $\mathbf{v}_i$ was not previously used by a registered user and that sig_i is a valid signature on $\mathbf{v}_i$ w.r.t. $\mathsf{upk}[i]$. It aborts if this is not the case. Otherwise, GM chooses a fresh ℓ-bit identifier $\mathrm{id}_i = \mathrm{id}_i[1] \ldots \mathrm{id}_i[\ell] \in \{0,1\}^\ell$ and uses $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ to certify $\mathcal{U}_i$ as a new group member. To this end, GM defines the matrix

$$\mathbf{A}_{\mathrm{id}_i} = \Big[\mathbf{A} \ \Big|\ \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \mathrm{id}_i[j]\mathbf{A}_j\Big] \in \mathbb{Z}_q^{n\times 2m}. \qquad (6)$$

Then, GM runs $\mathbf{T}'_{\mathrm{id}_i} \leftarrow \mathsf{ExtBasis}(\mathbf{A}_{\mathrm{id}_i}, \mathbf{T_A})$ to obtain a short delegated basis $\mathbf{T}'_{\mathrm{id}_i}$ of $\Lambda_q^{\perp}(\mathbf{A}_{\mathrm{id}_i}) \in \mathbb{Z}^{2m\times 2m}$. Finally, GM samples a short vector $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and uses the obtained delegated basis $\mathbf{T}'_{\mathrm{id}_i}$ to compute a short vector $\mathbf{d}_i = \begin{bmatrix}\mathbf{d}_{i,1} \\ \mathbf{d}_{i,2}\end{bmatrix} \in \mathbb{Z}^{2m}$ such that

$$\begin{aligned}\mathbf{A}_{\mathrm{id}_i}\mathbf{d}_i \cdot &= \Big[\mathbf{A} \ \Big|\ \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \mathrm{id}_i[j]\mathbf{A}_j\Big] \cdot \mathbf{d}_i \\ &= \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i\big) \bmod q. \qquad (7)\end{aligned}$$

The triple $(\mathrm{id}_i, \mathbf{d}_i, \mathbf{s}_i)$ is sent to $\mathcal{U}_i$. Then, $\mathsf{J}_{\mathsf{user}}$ verifies that the received $(\mathrm{id}_i, \mathbf{d}_i, \mathbf{s}_i)$ satisfies (7) and that $\|\mathbf{d}_i\|_\infty \leq \beta$, $\|\mathbf{s}_i\|_\infty \leq \beta$. If these conditions are not satisfies, $\mathsf{J}_{\mathsf{user}}$ aborts. Otherwise, $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate as $\mathsf{cert}_i = (\mathrm{id}_i, \mathbf{d}_i, \mathbf{s}_i)$. The membership secret sec_i is defined to be $\mathsf{sec}_i = \mathbf{z}_i \in \mathbb{Z}^{4m}$. J_{GM} stores $\mathsf{transcript}_i = (\mathbf{v}_i, \mathsf{cert}_i, i, \mathsf{upk}[i], sig_i)$ in the database St_{trans} of joining transcripts.

Sign$(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** To sign $M \in \{0,1\}^*$ using $\mathsf{cert}_i = (\mathrm{id}_i, \mathbf{d}_i, \mathbf{s}_i)$, where $\mathbf{d}_i = [\mathbf{d}_{i,1}^T \mid \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}^{2m}$ and $\mathbf{s}_i \in \mathbb{Z}^{2m}$, as well as the membership secret $\mathsf{sec}_i = \mathbf{z}_i \in \mathbb{Z}^{4m}$, the group member $\mathcal{U}_i$ generates a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(n)$ and conducts the following steps.

1. Compute $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n \times 2m}$ and use it as an IBE public key to encrypt $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{2m}$, where $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$ is the syndrome of $\mathsf{sec}_i = \mathbf{z}_i \in \mathbb{Z}^{4m}$ for the matrix $\mathbf{F}$. Namely, compute $\mathbf{c}_{\mathbf{v}_i} \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}$ as

$$\mathbf{c}_{\mathbf{v}_i} = (\mathbf{c}_1, \mathbf{c}_2) = \big(\mathbf{B}^T \cdot \mathbf{e}_0 + \mathbf{x}_1,\ \mathbf{G}_0^T \cdot \mathbf{e}_0 + \mathbf{x}_2 + \mathsf{bin}(\mathbf{v}_i) \cdot \lfloor q/2 \rfloor\big) \tag{8}$$

for randomly chosen $\mathbf{e}_0 \hookleftarrow \chi^n$, $\mathbf{x}_1 \hookleftarrow \chi^m, \mathbf{x}_2 \hookleftarrow \chi^{2m}$. Notice that, as in the construction of [62], the columns of $\mathbf{G}_0$ can be interpreted as public keys for the multi-bit version of the dual Regev encryption scheme.

2. Run the protocol in Section 5.5 to prove the knowledge of $\mathrm{id}_i \in \{0,1\}^\ell$, vectors $\mathbf{s}_i \in \mathbb{Z}^{2m}, \mathbf{d}_{i,1}, \mathbf{d}_{i,2} \in \mathbb{Z}^m, \mathbf{z}_i \in \mathbb{Z}^{4m}$ with infinity norm bound β; $\mathbf{e}_0 \in \mathbb{Z}^n$, $\mathbf{x}_1 \in \mathbb{Z}^m, \mathbf{x}_2 \in \mathbb{Z}^{2m}$ with infinity norm bound B and $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{2m}, \mathbf{w}_i \in \{0,1\}^m$, that satisfy (8) as well as

$$\mathbf{A} \cdot \mathbf{d}_{i,1} + \mathbf{A}_0 \cdot \mathbf{d}_{i,2} + \sum_{j=1}^{\ell} (\mathrm{id}_i[j] \cdot \mathbf{d}_{i,2}) \cdot \mathbf{A}_j - \mathbf{D} \cdot \mathbf{w}_i = \mathbf{u} \in \mathbb{Z}_q^n$$

and

$$\begin{cases} \mathbf{H}_{2n \times m} \cdot \mathbf{w}_i = \mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i \in \mathbb{Z}_q^{2n} \\ \mathbf{F} \cdot \mathbf{z}_i = \mathbf{H}_{4n \times 2m} \cdot \mathsf{bin}(\mathbf{v}_i) \in \mathbb{Z}_q^{4n}. \end{cases}$$

The protocol is repeated $t = \omega(\log n)$ times in parallel to achieve negligible soundness error, and then made non-interactive using the Fiat-Shamir heuristic [41] as a triple $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^t, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^t)$, where $\mathsf{Chall}_K = H(M, \mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \{\mathsf{Comm}_{K,j}\}_{j=1}^t) \in \{1,2,3\}^t$

4. Compute a one-time signature $sig = \mathcal{S}(\mathsf{SK}, (\mathbf{c}_{\mathbf{v}_i}, \pi_K))$.

Output the signature that consists of

$$\Sigma = \big(\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig\big). \tag{9}$$

Verify$(\mathcal{Y}, M, \Sigma)$**:** Parse Σ as in (9). Then, return 1 if and only if:

(i) $\mathcal{V}(\mathsf{VK}, (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{s}_i}, \mathbf{c}_{\mathrm{id}}, \pi_K), sig) = 1$;
(ii) The proof of knowledge π_K properly verifies.
Otherwise, return 0.

Open$(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \Sigma)$**:** Parse $\mathcal{S}_{\mathsf{OA}}$ as $\mathbf{T_B} \in \mathbb{Z}^{m \times m}$ and Σ as in (9).

1. Compute $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n \times 2m}$. Then, using $\mathbf{T_B}$ to compute a small-norm matrix $\mathbf{E}_{0,\mathsf{VK}} \in \mathbb{Z}^{m \times 2m}$ such that $\mathbf{B} \cdot \mathbf{E}_{0,\mathsf{VK}} = \mathbf{G}_0 \bmod q$.
2. Using $\mathbf{E}_{0,\mathsf{VK}}$, decrypt $\mathbf{c}_{\mathbf{v}_i}$ to obtain a string $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{2m}$ (i.e., by computing $\lfloor (\mathbf{c}_2 - \mathbf{E}_{0,\mathsf{VK}}^T \cdot \mathbf{c}_1)/(q/2) \rceil$).
3. Determine if the $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{2m}$ obtained at step 2 corresponds to a vector $\mathbf{v} = \mathbf{H}_{4n \times 2m} \cdot \mathsf{bin}(\mathbf{v}) \bmod q$ that appears in a record $\mathsf{transcript}_i = (\mathbf{v}, \mathsf{cert}_i, i, \mathsf{upk}[i], sig_i)$ of the database St_{trans} for some i. If so, output the corresponding i (and, optionally, $\mathsf{upk}[i]$). Otherwise, output $\perp$.

We remark that the scheme readily extends to provide a mechanism whereby the opening authority can efficiently prove that signatures were correctly opened at each opening operation. The difference between the dynamic group signature models suggested by Kiayias and Yung [56] and Bellare *et al.* [13] is that, in the latter, the opening authority (OA) must be able to convince a judge that the Open algorithm was run correctly. Here, such a mechanism can be realized using the techniques of public-key encryption with non-interactive opening [38]. Namely, since $\mathsf{bin}(\mathbf{v}_i)$ is encrypted using an IBE scheme for the identity VK, the OA can simply reveal the decryption matrix $\mathbf{E}_{0,\mathsf{VK}}$, that satisfies $\mathbf{B} \cdot \mathbf{E}_{0,\mathsf{VK}} = \mathbf{G}_0 \bmod q$ (which corresponds to the verification of a GPV signature) and allows the verifier to perform step 2 of the opening algorithm himself. The resulting construction is easily seen to satisfy the notion of opening soundness of Sakai *et al.* [71].

4.2 Security Analysis

Due to the fact that the number of public matrices $\{\mathbf{A}_j\}_{j=0}^{\ell}$ is only logarithmic in $N_{\mathsf{gs}} = 2^{\ell}$ instead of being linear in the security parameter λ, the proof of security against misidentification attacks (as defined in Appendix A) cannot rely on the security of our signature scheme in a modular manner. The reason is that, at each run of the Join protocol, the group manager maintains a state and, instead of choosing the ℓ-bit identifier id uniformly in $\{0,1\}^{\ell}$, it chooses an identifier that has not been used yet. Since $\ell \ll \lambda$ (given that $N_{\mathsf{gs}} = 2^{\ell}$ is polynomial in λ), we thus have to prove security from scratch. However, the strategy of the reduction is exactly the same as in the security proof of the signature scheme.

Theorem 4. *The scheme is secure against misidentification attacks under the* $\mathsf{SIS}_{n,2m,q,\beta'}$ *assumption, where* $\beta' = \mathcal{O}(\ell\sigma^2 m^{3/2})$. (The proof is available in Appendix C.1.)

Theorem 5. *The scheme is secure against framing attacks under the* $\mathsf{SIS}_{4n,4m,q,\beta''}$ *assumption, where* $\beta'' = 4\sigma\sqrt{m}$. (The proof is given in Appendix C.2.)

Theorem 6. *Suppose that* Π^{OTS} *is a strongly unforgeable one-time signature. In the random oracle model, the scheme provides CCA-anonymity under the* $\mathsf{LWE}_{n,q,\chi}$ *assumption. Namely, for any* PPT *adversary* $\mathcal{A}$ *with advantage* ε*, there exists an algorithm* $\mathcal{B}$ *solving the* $\mathsf{LWE}_{n,q,\chi}$ *problem with advantage at most* $2^{-\Omega(n)}$ *smaller.* (The proof is given in Appendix C.3.)

5 Supporting Zero-Knowledge Argument Systems

This section provides a general framework that allows obtaining zero-knowledge arguments of knowledge (ZKAoK) for many relations appearing in lattice-based cryptography. Since lattice-based cryptosystems are built upon the hardness of the SIS and LWE problems, the relations among objects of the schemes are typically represented by modular linear equations. Thanks to the linearity property, we can often unify the given equations into one equation of the form:

$$\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q, \tag{10}$$

where $(\mathbf{P}, \mathbf{v})$ are public and $\mathbf{x}$ is a secret vector (or matrix) that possesses some constraints to be proven in zero-knowledge, e.g., its smallness (like a SIS solution or an LWE noise) or a special arrangement of its entries. Starting from this high-level observation, we look for a tool that handles these constraints well.

Stern's protocol [72], originally proposed in the context of code-based cryptography, appears to be well-suited for our purpose. Stern's main idea is simple, yet elegant: To prove that a binary vector $\mathbf{x}$ has the fixed-Hamming-weight constraint, simply send the verifier a random permutation $\pi(\mathbf{x})$ which should guarantee that the constraint is satisfied while leaking no additional information about $\mathbf{x}$. Ling *et al.* [61] developed this idea to handle the smallness constraint, via a technique called Decomposition-Extension. This technique decomposes a vector with small infinity norm $B \geq 1$ into $\lfloor \log_2 B \rfloor + 1$ vectors with infinity norm 1, and then, extends these vectors into elements of sets that are closed under permutations. Several subsequent works [58][62][60] employed the techniques of [72,61] in different contexts, but did not address the applicability and flexibility of the protocol in an abstract, generalized manner.

In Section 5.1, we abstract Stern's protocol to capture many relations that naturally appear in lattice-based cryptography. In particular, the argument systems used in our signature with efficient protocols (Section 3) and dynamic group signature (Section 4) can all be derived from this abstract protocol, which we will demonstrate in Sections 5.3, 5.4 and 5.5, respectively.

We note that several works [52,73,15] addressed the problem of proving multiplicative and additive relations among committed linear objects (matrices and vectors over $\mathbb{Z}_q$) in lattice-based cryptography. These results, however, do not yield a simple solution for the relations involved in our schemes. If we were to plug proof systems like [52,73,15] in our relations, we would need to commit to all objects using perfectly binding commitments (which would require very long commitment keys) and express the relations in terms of many multiplications and additions gates before running many instances of the proof systems depending on the circuit. Instead of considering general circuits, our framework aims at a more direct (but still fairly general) solution for a large class of relations that naturally appear in SIS and LWE-based cryptography.

5.1 Abstracting Stern's Protocol

Let $D, L, q \geq 2$ be positive integers let VALID be a subset of $\{-1, 0, 1\}^L$. Suppose that $\mathcal{S}$ is a finite set such that one can associate every $\pi \in \mathcal{S}$ with a permutation

T_π of L elements, satisfying the following condition:

$$\mathbf{x} \in \mathsf{VALID} \iff T_\pi(\mathbf{x}) \in \mathsf{VALID}. \tag{11}$$

We aim to construct a statistical ZKAoK for the following abstract relation:

$$\mathrm{R}_{\mathrm{abstract}} = \big\{(\mathbf{P}, \mathbf{v}), \mathbf{x} \in \mathbb{Z}_q^{D \times L} \times \mathbb{Z}_q^{D} \times \mathsf{VALID} : \mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q.\big\}$$

Note that, Stern's original protocol corresponds to the special case when $\mathsf{VALID} = \{\mathbf{x} \in \{0,1\}^L : \mathsf{wt}(\mathbf{x}) = k\}$ (where $\mathsf{wt}(\cdot)$ denotes the Hamming weight and $k < L$ is a given integer), $\mathcal{S} = \mathcal{S}_L$ - hereunder the set of all permutations of L elements, and $T_\pi(\mathbf{x}) = \pi(\mathbf{x})$.

The equivalence in (11) plays a crucial role in proving in ZK that $\mathbf{x} \in \mathsf{VALID}$: To do so, the prover samples $\pi \hookleftarrow U(\mathcal{S})$ and let the verifier check that $T_\pi(\mathbf{x}) \in \mathsf{VALID}$, while the latter cannot learn any additional information about $\mathbf{x}$ thanks to the randomness of π. Furthermore, to prove in ZK that the linear equation holds, the prover samples a masking vector $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, sends $\mathbf{y} = \mathbf{x} + \mathbf{r} \bmod q$, and convinces the verifier instead that $\mathbf{P} \cdot \mathbf{y} = \mathbf{P} \cdot \mathbf{r} + \mathbf{v} \bmod q$.

The interactive protocol between the prover and the verifier with common input $(\mathbf{P}, \mathbf{v})$ and prover's secret input $\mathbf{x}$ is described in Figure 1. The protocol employs a statistically hiding and computationally binding string commitment scheme COM (e.g., the SIS-based one from [53]).

1. **Commitment:** Prover samples $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$ and randomness ρ_1, ρ_2, ρ_3 for COM. Then he sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to the verifier, where

   $$C_1 = \mathsf{COM}(\pi, \mathbf{P} \cdot \mathbf{r}; \rho_1), \;\; C_2 = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2), \;\; C_3 = \mathsf{COM}(T_\pi(\mathbf{x} + \mathbf{r}); \rho_3).$$

2. **Challenge:** The verifier sends a challenge $Ch \hookleftarrow U(\{1,2,3\})$ to the prover.
3. **Response:** Depending on Ch, the prover sends RSP computed as follows:
   - $Ch = 1$: Let $\mathbf{t}_x = T_\pi(\mathbf{x})$, $\mathbf{t}_r = T_\pi(\mathbf{r})$, and $\mathrm{RSP} = (\mathbf{t}_x, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\pi_2 = \pi$, $\mathbf{y} = \mathbf{x} + \mathbf{r}$, and $\mathrm{RSP} = (\pi_2, \mathbf{y}, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\pi_3 = \pi$, $\mathbf{r}_3 = \mathbf{r}$, and $\mathrm{RSP} = (\pi_3, \mathbf{r}_3, \rho_1, \rho_2)$.

Verification: Receiving RSP, the verifier proceeds as follows:

- $Ch = 1$: Check that $\mathbf{t}_x \in \mathsf{VALID}$ and $C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2)$, $C_3 = \mathsf{COM}(\mathbf{t}_x + \mathbf{t}_r; \rho_3)$.
- $Ch = 2$: Check that $C_1 = \mathsf{COM}(\pi_2, \mathbf{P} \cdot \mathbf{y} - \mathbf{v}; \rho_1)$, $C_3 = \mathsf{COM}(T_{\pi_2}(\mathbf{y}); \rho_3)$.
- $Ch = 3$: Check that $C_1 = \mathsf{COM}(\pi_3, \mathbf{P} \cdot \mathbf{r}_3; \rho_1)$, $C_2 = \mathsf{COM}(T_{\pi_3}(\mathbf{r}_3); \rho_2)$.

In each case, the verifier outputs 1 if and only if all the conditions hold.

Fig. 1: A ZKAoK for the relation $\mathrm{R}_{\mathrm{abstract}}$.

The properties of the given protocol are summarized in the following lemma.

Lemma 8. *The protocol in Figure 1 is a statistical* ZKAoK *for the relation* $\mathrm{R}_{\mathrm{abstract}}$ *with perfect completeness, soundness error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(L \log q)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{P}, \mathbf{v})$*, outputs an accepted transcript which is statistically close to that produced by the real prover.*
- *There exists an efficient knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* Ch*, outputs* $\mathbf{x}' \in \mathsf{VALID}$ *such that* $\mathbf{P} \cdot \mathbf{x}' = \mathbf{v} \bmod q$.

The proof of Lemma 8 employs standard simulation and extraction techniques for Stern-like protocols [53][61][62]. We defer it to Appendix D.

5.2 Supporting Notations and Techniques

Below we will describe the notations and techniques, adapted from recent works on Stern-like protocols [61][58][40][60], that we will employ in the next subsections to handle 3 different constraints of the witness vectors.

Let m be an arbitrary dimension, and B be an arbitrary infinity norm bound.

Case 1: $\mathbf{w} \in \{0,1\}^m$. We denote by B^2_m the set of all vectors in $\{0,1\}^{2m}$ having exactly m coordinates equal to 1. We also let $\mathsf{Ext}_{2m}(\mathbf{w})$ be the algorithm that outputs a vector $\hat{\mathbf{w}} \in \mathsf{B}^2_m$ by appending m suitable coordinates to $\mathbf{w} \in \{0,1\}^m$. Note that, for any permutation $\rho \in \mathcal{S}_{2m}$, we have $\hat{\mathbf{w}} \in \mathsf{B}^2_m \Leftrightarrow \rho(\hat{\mathbf{w}}) \in \mathsf{B}^2_m$.

Case 2: $\mathbf{w} \in [-B, B]^m$. We define the number $\delta_B := \lfloor \log_2 B \rfloor + 1$, and denote by $\mathsf{B}^3_{m\delta_B}$ the set of all vectors in $\{-1,0,1\}^{3m\delta_B}$ having exactly $m\delta_B$ coordinates equal to j, for every $j \in \{-1,0,1\}$. The Decomposition-Extension technique from [61] consists of transforming $\mathbf{w} \in [-B,B]^m$ to a vector $\mathsf{DecExt}_{m,B}(\mathbf{w}) \in \mathsf{B}^3_{m\delta_B}$, as follows.

Define the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \left\lfloor \frac{B+2^{j-1}}{2^j} \right\rfloor$ for all $j \in [1, \delta_B]$. As noted in [61], this sequence satisfies $\sum_{j=1}^{\delta_B} B_j = B$, and for any $w \in [-B, B]$, one can efficiently compute $w^{(1)}, \ldots, w^{(\delta_B)} \in \{-1,0,1\}$ such that $\sum_{j=1}^{\delta_B} B_j \cdot w^{(j)} = w$. Next, define the matrix

$$\mathbf{K}_{m,B} = \mathbf{I}_m \otimes [B_1 \,|\, \ldots \,|\, B_{\delta_B}] = \begin{bmatrix} B_1 \ldots B_{\delta_B} & & \\ & \ddots & \\ & & B_1 \ldots B_{\delta_B} \end{bmatrix} \in \mathbb{Z}^{m \times m\delta_B},$$

and its extension $\widehat{\mathbf{K}}_{m,B} = \left[\mathbf{K}_{m,B} \middle| \mathbf{0}^{m \times 2m\delta_B}\right] \in \mathbb{Z}^{m \times 3m\delta_B}$.

If we let $\mathbf{w} = (w_1, \ldots, w_m)^T$, then we can compute

$$\mathbf{w}' = \left(w_1^{(1)}, \ldots, w_1^{(\delta_B)}, \ldots, w_m^{(1)}, \ldots, w_m^{(\delta_B)}\right)^T \in \{-1,0,1\}^{m\delta_B}$$

satisfying $\mathbf{K}_{m,B} \cdot \mathbf{w}' = \mathbf{w}$. By appending $2m\delta_B$ suitable coordinates to $\mathbf{w}'$, we can obtain $\hat{\mathbf{w}} \in \mathsf{B}^3_{m\delta_B}$ satisfying $\widehat{\mathbf{K}}_{m,B} \cdot \hat{\mathbf{w}} = \mathbf{w}$.

Note that for any $\phi \in \mathcal{S}_{3m\delta_B}$, we have $\hat{\mathbf{w}} \in \mathsf{B}^3_{m\delta_B} \Leftrightarrow \phi(\hat{\mathbf{w}}) \in \mathsf{B}^3_{m\delta_B}$.

Case 3: $\mathbf{w} \in \{0,1\}^{2m}$ is the correct encoding of some $\mathbf{t} \in \{0,1\}^m$.

Recall that the encoding function from Section 3.3, hereunder denoted by Encode_m if the input is a binary vector of length m, extends $\mathbf{t} = (t_1, \ldots, t_m)^T$

to $\mathsf{Encode}_m(\mathbf{t}) = (\bar{t}_1, t_1, \ldots, \bar{t}_m, t_m)$. We define $\mathsf{CorEnc}(m) = \{\mathbf{w} = \mathsf{Encode}_m(\mathbf{t}) : \mathbf{t} \in \{0,1\}^m\}$ - the set of all correct encodings of m-bit vectors. To handle the constraint $\mathbf{w} \in \mathsf{CorEnc}(m)$, we adapt the permuting technique from [58][40][60].

For $\mathbf{b} = (b_1, \ldots, b_m)^T \in \{0,1\}^m$, we let $E_{\mathbf{b}}$ be the permutation transforming vector $\mathbf{w} = (w_1^0, w_1^1, \ldots, w_m^0, w_m^1) \in \mathbb{Z}^{2m}$ to $E_{\mathbf{b}}(\mathbf{w}) = (w_1^{b_1}, w_1^{\bar{b}_1}, \ldots, w_m^{b_m}, w_m^{\bar{b}_m})$. Note that, $E_{\mathbf{b}}$ transforms $\mathbf{w} = \mathsf{Encode}_m(\mathbf{t})$ to $E_{\mathbf{b}}(\mathbf{w}) = \mathsf{Encode}_m(\mathbf{t} \oplus \mathbf{b})$, where $\oplus$ denotes the bit-wise addition modulo 2. Thus, for any $\mathbf{b} \in \{0,1\}^m$, we have

$$\mathbf{w} \in \mathsf{CorEnc}(m) \Leftrightarrow E_{\mathbf{b}}(\mathbf{w}) \in \mathsf{CorEnc}(m).$$

5.3 Proving the Consistency of Commitments

The argument system used in our protocol for signing a committed value in Section 3.3 can be summarized as follows.

Common Input: Matrices $\{\mathbf{D}_k \in \mathbb{Z}_q^{2n \times 2m}\}_{k=0}^N$; $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$; $\mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m}$;
vectors $\mathbf{c}_{\mathfrak{m}} \in \mathbb{Z}_q^{2n}$; $\{\mathbf{c}_{k,1} \in \mathbb{Z}_q^m\}_{k=1}^N$; $\{\mathbf{c}_{k,2} \in \mathbb{Z}_q^{2m}\}_{k=1}^N$; $\mathbf{c}_{\mathbf{s}',1} \in \mathbb{Z}_q^m$; $\mathbf{c}_{\mathbf{s}',2} \in \mathbb{Z}_q^{2m}$.

Prover's Input: $\mathfrak{m} = (\mathfrak{m}_1^T \| \ldots \| \mathfrak{m}_N^T)^T \in \mathsf{CorEnc}(mN)$;
$\{\mathbf{s}_{k,0} \in [-B,B]^n,\ \mathbf{e}_{k,1} \in [-B,B]^m;\ \mathbf{e}_{k,2} \in [-B,B]^{2m}\}_{k=1}^N$; $\mathbf{s}_0 \in [-B,B]^n$;
$\mathbf{e}_{0,1} \in [-B,B]^m$; $\mathbf{e}_{0,2} \in [-B,B]^{2m}$; $\mathbf{s}' \in [-(p-1),(p-1)]^{2m}$

Prover's Goal: Convince the verifier in ZK that:

$$\begin{cases} \mathbf{c}_{\mathfrak{m}} = \mathbf{D}_0 \cdot \mathbf{s}' + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k \bmod q; \\ \mathbf{c}_{\mathbf{s}',1} = \mathbf{B}^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,1} \bmod q;\ \ \mathbf{c}_{\mathbf{s}',2} = \mathbf{G}_1^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,2} + \lfloor q/p \rfloor \cdot \mathbf{s}' \bmod q; \\ \forall k \in [N] : \mathbf{c}_{k,1} = \mathbf{B}^T \cdot \mathbf{s}_{k,0} + \mathbf{e}_{k,1};\ \ \mathbf{c}_{k,2} = \mathbf{G}_1^T \cdot \mathbf{s}_{k,0} + \mathbf{e}_{k,2} + \lfloor q/2 \rfloor \cdot \mathfrak{m}_k. \end{cases} \tag{12}$$

We will show that the above argument system can be obtained from the one in Section 5.1. We proceed in 2 steps.

Step 1: *Transforming the equations in (12) into a unified one of the form* $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$*, where* $\|\mathbf{x}\|_\infty = 1$ *and* $\mathbf{x} \in \mathsf{VALID}$ *- a "specially-designed" set.*

To do so, we first form the following vectors and matrices:

$$\begin{cases} \mathbf{x}_1 = \left(\mathbf{s}_0^T \| \mathbf{e}_{0,1}^T \| \mathbf{e}_{0,2}^T \| \mathbf{s}_{1,0}^T \| \mathbf{e}_{1,1}^T \| \mathbf{e}_{1,2}^T \| \ldots \| \mathbf{s}_{N,0}^T \| \mathbf{e}_{N,1}^T \| \mathbf{e}_{N,2}^T\right)^T \in [-B,B]^{(n+3m)(N+1)}; \\ \mathbf{v} = \left(\mathbf{c}_{\mathfrak{m}}^T \| \mathbf{c}_{\mathbf{s}',1}^T \| \mathbf{c}_{\mathbf{s}',2}^T \| \mathbf{c}_{1,1}^T \| \mathbf{c}_{1,2}^T \| \ldots \| \mathbf{c}_{N,1}^T \| \mathbf{c}_{N,2}^T\right)^T \in \mathbb{Z}_q^{2n+3m(N+1)}; \\ \mathbf{P}_1 = \left(\begin{array}{c|c} \mathbf{B}^T & \\ \hline \mathbf{G}_1^T & \end{array} \mathbf{I}_{3m}\right); \quad \mathbf{Q}_2 = \left(\begin{array}{c} \mathbf{0} \\ \hline \lfloor \frac{q}{2} \rfloor \mathbf{I}_{2m} \end{array}\right); \quad \mathbf{Q}_p = \left(\begin{array}{c} \mathbf{0} \\ \hline \lfloor \frac{q}{p} \rfloor \mathbf{I}_{2m} \end{array}\right) \\ \mathbf{M}_1 = \left(\begin{array}{cccc} & \mathbf{0} & & \\ \hline \mathbf{P}_1 & & & \\ & \mathbf{P}_1 & & \\ & & \ddots & \\ & & & \mathbf{P}_1 \end{array}\right); \quad \mathbf{M}_2 = \left(\begin{array}{ccc} \mathbf{D}_1 | \ldots | \mathbf{D}_N & & \\ \hline & \mathbf{0} & \\ \hline \mathbf{Q}_2 & & \\ & \ddots & \\ & & \mathbf{Q}_2 \end{array}\right); \quad \mathbf{M}_3 = \left(\begin{array}{c} \mathbf{D}_0 \\ \hline \mathbf{Q}_p \\ \hline \\ \mathbf{0} \\ \\ \end{array}\right). \end{cases}$$

We then observe that (12) can be rewritten as:

$$\mathbf{M}_1 \cdot \mathbf{x}_1 + \mathbf{M}_2 \cdot \mathfrak{m} + \mathbf{M}_3 \cdot \mathbf{s}' = \mathbf{v} \in \mathbb{Z}_q^D, \tag{13}$$

where $D = 2n + 3m(N+1)$. Now we employ the techniques from Section 5.2 to convert (13) into the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$. Specifically, if we let:

$$\begin{cases} \mathsf{DecExt}_{(n+3m)(N+1),B}(\mathbf{x}_1) \to \hat{\mathbf{x}}_1 \in \mathsf{B}^3_{(n+3m)(N+1)\delta_B}; \\ \mathbf{M}_1' = \mathbf{M}_1 \cdot \widehat{\mathbf{K}}_{(n+3m)(N+1),B} \in \mathbb{Z}_q^{D \times 3(n+3m)(N+1)\delta_B}; \\ \mathsf{DecExt}_{2m,p-1}(\mathbf{s}') \to \hat{\mathbf{s}} \in \mathsf{B}^3_{2m\delta_{p-1}}; \;\; \mathbf{M}_3' = \mathbf{M}_3 \cdot \widehat{\mathbf{K}}_{2m,p-1} \in \mathbb{Z}_q^{D \times 6m\delta_{p-1}}, \end{cases}$$

$L = 3(n+3m)(N+1)\delta_B + 2mN + 6m\delta_{p-1}$, and $\mathbf{P} = \left[\mathbf{M}_1' | \mathbf{M}_2 | \mathbf{M}_3'\right] \in \mathbb{Z}_q^{D \times L}$, and $\mathbf{x} = \left(\hat{\mathbf{x}}_1^T \| \mathfrak{m}^T \| \hat{\mathbf{s}}^T\right)^T$, then we will obtain the desired equation:

$$\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q.$$

Having performed the above unification, we now define VALID as the set of all vectors $\mathbf{t} \in \{-1,0,1\}^L$ of the form $\mathbf{t} = \left(\mathbf{t}_1^T \| \mathbf{t}_2^T \| \mathbf{t}_3^T\right)^T$, where $\mathbf{t}_1 \in \mathsf{B}^3_{(n+3m)(N+1)\delta_B}$, $\mathbf{t}_2 \in \mathsf{CorEnc}(mN)$, and $\mathbf{t}_3 \in \mathsf{B}^3_{2m\delta_{p-1}}$. Note that $\mathbf{x} \in \mathsf{VALID}$.

Step 2: *Specifying the set $\mathcal{S}$ and permutations of L elements $\{T_\pi : \pi \in \mathcal{S}\}$ for which the equivalence (11) holds.*

- Define $\mathcal{S} := \mathcal{S}_{3(n+3m)(N+1)\delta_B} \times \{0,1\}^{mN} \times \mathcal{S}_{6m\delta_{p-1}}$.
- For $\pi = (\pi_1, \mathbf{b}, \pi_3) \in \mathcal{S}$, and for vector $\mathbf{w} = \left(\mathbf{w}_1^T \| \mathbf{w}_2^T \| \mathbf{w}_3^T\right)^T \in \mathbb{Z}_q^L$, where $\mathbf{w}_1 \in \mathbb{Z}_q^{3(n+3m)(N+1)\delta_B}$, $\mathbf{w}_2 \in \mathbb{Z}_q^{2mN}$, $\mathbf{w}_3 \in \mathbb{Z}_q^{6m\delta_{p-1}}$, we define:

$$T_\pi = \left(\pi_1(\mathbf{w}_1)^T \| E_{\mathbf{b}}(\mathbf{w}_2)^T \| \pi_3(\mathbf{w}_3)^T\right)^T.$$

By inspection, it can be seen that the property (11) is satisfied, as desired. As a result, we can obtain the required argument system by running the protocol in Section 5.1 with common input $(\mathbf{P}, \mathbf{v})$ and prover's input $\mathbf{x}$.

5.4 Proving the Possession of a Signature on a Committed Value

We now describe how to derive the protocol for proving the possession of a signature on a committed value, that is used in Section 3.3.

Common Input: Matrices $\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D} \in \mathbb{Z}_q^{n \times m}$; $\{\mathbf{D}_k \in \mathbb{Z}_q^{2n \times 2m}\}_{k=0}^N$; $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$; $\mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m}$; $\mathbf{G}_0 \in \mathbb{Z}_q^{n \times \ell}$; vectors $\mathbf{c}_{\tau,1}, \{\mathbf{c}_{k,1}\}_{k=1}^N, \mathbf{c}_{\mathbf{v}_1,1}, \mathbf{c}_{\mathbf{v}_2,1}, \mathbf{c}_{s,1} \in \mathbb{Z}_q^m$; $\mathbf{c}_{\tau,2} \in \mathbb{Z}_q^\ell$; $\{\mathbf{c}_{k,2}\}_{k=1}^N, \mathbf{c}_{s,2} \in \mathbb{Z}_q^{2m}$; $\mathbf{c}_{\mathbf{v}_1,2}, \mathbf{c}_{\mathbf{v}_2,2} \in \mathbb{Z}_q^m$; $\mathbf{u} \in \mathbb{Z}_q^n$.

Prover's Input: $\tau \in \{0,1\}^\ell$; $\mathbf{v}_1, \mathbf{v}_2 \in [-\beta, \beta]^m$ - where $\beta = \sigma \cdot \omega(\log m)$ - the infinity norm bound of signatures; $\mathbf{s} \in [-(p-1), (p-1)]^{2m}$;
$\mathfrak{m} = (\mathfrak{m}_1^T \| \ldots \| \mathfrak{m}_N^T)^T \in \mathsf{CorEnc}(mN)$; $\{\mathbf{s}_{k,0}\}_{k=1}^N$, $\mathbf{s}_{\mathbf{v}_1}$, $\mathbf{s}_{\mathbf{v}_2}$, $\mathbf{s}_0$, $\mathbf{s}_\tau \in [-B, B]^n$;
$\{\mathbf{e}_{k,1}\}_{k=1}^N$, $\mathbf{e}_{\mathbf{v}_1,1}$, $\mathbf{e}_{\mathbf{v}_2,1}$, $\mathbf{e}_{0,1}$, $\mathbf{e}_{\tau,1} \in [-B, B]^m$; $\{\mathbf{e}_{k,2}\}_{k=1}^N, \mathbf{e}_{0,2} \in [-B, B]^{2m}$;
$\mathbf{e}_{\mathbf{v}_1,2}$, $\mathbf{e}_{\mathbf{v}_2,2} \in [-B, B]^m$; $\mathbf{e}_{\tau,2} \in [-B, B]^\ell$.

Prover's Goal: Convince the verifier in ZK that:

$$\mathbf{A}\cdot\mathbf{v}_1 + \mathbf{A}_0\cdot\mathbf{v}_2 + \sum_{i=1}^{\ell}\mathbf{A}_i\cdot\tau[i]\mathbf{v}_2 - \mathbf{D}\cdot\mathsf{bin}(\mathbf{D}_0\cdot\mathbf{s} + \sum_{k=1}^{N}\mathbf{D}_i\cdot\mathfrak{m}_k) = \mathbf{u} \bmod q, \quad (14)$$

and that

$$\begin{cases} \forall k\in[N] : \mathbf{c}_{k,1} = \mathbf{B}^T\cdot\mathbf{s}_{k,0} + \mathbf{e}_{k,1};\;\; \mathbf{c}_{k,2} = \mathbf{G}_1^T\cdot\mathbf{s}_{k,0} + \mathbf{e}_{k,2} + \lfloor q/2\rfloor\cdot\mathfrak{m}_k; \\ \mathbf{c}_{\mathbf{v}_1,1} = \mathbf{B}^T\cdot\mathbf{s}_{\mathbf{v}_1} + \mathbf{e}_{\mathbf{v}_1,1} \bmod q;\; \mathbf{c}_{\mathbf{v}_1,2} = \mathbf{G}_1^T\cdot\mathbf{s}_{\mathbf{v}_1} + \mathbf{e}_{\mathbf{v}_1,2} + \lfloor \frac{q}{p}\rfloor\cdot\mathbf{v}_1 \bmod q; \\ \mathbf{c}_{\mathbf{v}_2,1} = \mathbf{B}^T\cdot\mathbf{s}_{\mathbf{v}_2} + \mathbf{e}_{\mathbf{v}_2,1} \bmod q;\; \mathbf{c}_{\mathbf{v}_2,2} = \mathbf{G}_1^T\cdot\mathbf{s}_{\mathbf{v}_2} + \mathbf{e}_{\mathbf{v}_2,2} + \lfloor \frac{q}{p}\rfloor\cdot\mathbf{v}_2 \bmod q; \\ \mathbf{c}_{\mathbf{s},1} = \mathbf{B}^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,1} \bmod q;\;\; \mathbf{c}_{\mathbf{s},2} = \mathbf{G}_1^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,2} + \lfloor q/p\rfloor\cdot\mathbf{s} \bmod q; \\ \mathbf{c}_{\tau,1} = \mathbf{B}^T\cdot\mathbf{s}_\tau + \mathbf{e}_{\tau,1} \bmod q;\; \mathbf{c}_{\tau,2} = \mathbf{G}_0^T\cdot\mathbf{s}_\tau + \mathbf{e}_{\tau,2} + \lfloor q/2\rfloor\cdot\tau \bmod q. \end{cases} \quad (15)$$

We proceed in 2 steps.

Step 1: *Transforming the equations in (14) and 15 into a unified one of the form* $\mathbf{P}\cdot\mathbf{x} = \mathbf{v} \bmod q$, *where* $\|\mathbf{x}\|_\infty = 1$ *and* $\mathbf{x}\in\mathsf{VALID}$ *- a "specially-designed" set.*

Note that, if we let $\mathbf{y} = \mathsf{bin}(\mathbf{D}_0\cdot\mathbf{s} + \sum_{k=1}^{N}\mathbf{D}_i\cdot\mathfrak{m}_k) \in \{0,1\}^m$, then we have $\mathbf{H}_{2n\times m}\cdot\mathbf{y} = \mathbf{D}_0\cdot\mathbf{s} + \sum_{k=1}^{N}\mathbf{D}_i\cdot\mathfrak{m}_k \bmod q$ and (14) can be equivalently written as:

$$\begin{pmatrix}\mathbf{A}\\ \mathbf{0}\end{pmatrix}\cdot\mathbf{v}_1 + \begin{pmatrix}\mathbf{A}_0\\ \mathbf{0}\end{pmatrix}\cdot\mathbf{v}_2 + \sum_{i=1}^{\ell}\begin{pmatrix}\mathbf{A}_i\\ \mathbf{0}\end{pmatrix}\cdot\tau[i]\mathbf{v}_2 + \begin{pmatrix}\mathbf{0}\\ \mathbf{D}_0\end{pmatrix}\cdot\mathbf{s} + \begin{pmatrix}-\mathbf{D}\\ -\mathbf{H}_{2n\times m}\end{pmatrix}\cdot\mathbf{y}$$
$$+ \begin{pmatrix}\mathbf{0}\\ \mathbf{D}_1|\ldots|\mathbf{D}_N\end{pmatrix}\cdot\mathfrak{m} = \begin{pmatrix}\mathbf{u}\\ \mathbf{0}^{2n}\end{pmatrix} \bmod q.$$

Next, we use basic linear algebra to combine this equation and (15) into:

$$\mathbf{F}\cdot\mathbf{v}_1 + \mathbf{F}_0\cdot\mathbf{v}_2 + \sum_{i=1}^{\ell}\mathbf{F}_i\cdot\tau[i]\mathbf{v}_2 + \mathbf{M}_1\cdot\tau + \mathbf{M}_2\cdot\mathbf{y} + \mathbf{M}_3\cdot\mathfrak{m} + \mathbf{M}_4\cdot\mathbf{s} = \mathbf{v} \bmod q, \quad (16)$$

where matrices $\mathbf{F}, \mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_\ell \in \mathbb{Z}_q^{D\times m}$, $\mathbf{M}_1 \in \mathbb{Z}_q^{D\times\ell}$, $\mathbf{M}_2 \in \mathbb{Z}_q^{D\times m}$, $\mathbf{M}_3 \in \mathbb{Z}_q^{D\times 2mN}$, $\mathbf{M}_4 \in \mathbb{Z}_q^{D\times 2m}$, and vector $\mathbf{v}\in\mathbb{Z}_q^D$ are built from the public input, for dimension $D = \ell + 3n + 8m + 3mN$.

Now we further transform (16) using the techniques from Section 5.2. Specifically, we form the following:

$$\begin{cases} \mathsf{DecExt}_{m,\beta}(\mathbf{v}_1) \to \hat{\mathbf{v}}_1 \in \mathsf{B}^3_{m\delta_\beta};\;\; \mathsf{DecExt}_{m,\beta}(\mathbf{v}_2) \to \hat{\mathbf{v}}_2 \in \mathsf{B}^3_{m\delta_\beta}; \\ \mathbf{F}' = \left[\mathbf{F}\cdot\widehat{\mathbf{K}}_{m,\beta}|\mathbf{F}_0\cdot\widehat{\mathbf{K}}_{m,\beta}|\mathbf{F}_1\cdot\widehat{\mathbf{K}}_{m,\beta}|\ldots|\mathbf{F}_\ell\cdot\widehat{\mathbf{K}}_{m,\beta}|\mathbf{0}^{D\times 3m\delta_\beta\ell}\right] \in \mathbb{Z}_q^{D\times 3m\delta_\beta(2\ell+2)}; \\ \mathsf{Ext}_{2\ell}(\tau) \to \hat{\tau} = (\tau[1],\ldots,\tau[\ell],\ldots,\tau[2\ell])^T \in \mathsf{B}^2_\ell;\; \mathbf{M}'_1 = [\mathbf{M}_1|\mathbf{0}^{D\times\ell}] \in \mathbb{Z}_q^{D\times 2\ell}; \\ \mathsf{Ext}_{2m}(\mathbf{y}) \to \hat{\mathbf{y}} \in \mathsf{B}^2_m;\; \mathbf{M}'_2 = [\mathbf{M}_2|\mathbf{0}^{D\times m}] \in \mathbb{Z}_q^{D\times 2m}; \\ \mathsf{DecExt}_{2m,p-1}(\mathbf{s}) \to \hat{\mathbf{s}} \in \mathsf{B}^3_{2m\delta_{p-1}};\; \mathbf{M}'_4 = \mathbf{M}_4\cdot\widehat{\mathbf{K}}_{2m,p-1} \in \mathbb{Z}_q^{D\times 6m\delta_{p-1}}. \end{cases}$$

Now, let $L = 3m\delta_\beta(2\ell+2) + 2\ell + 2m + 2mN + 6m\delta_{p-1}$, and construct matrix $\mathbf{P} = \left[\mathbf{F}'|\mathbf{M}'_1|\mathbf{M}'_2|\mathbf{M}_3|\mathbf{M}'_4\right] \in \mathbb{Z}_q^{D\times L}$ and vector

$$\mathbf{x} = \left(\hat{\mathbf{v}}_1^T \,\|\, \hat{\mathbf{v}}_2^T \,\|\, \tau[1]\hat{\mathbf{v}}_2^T \,\|\, \ldots \,\|\, \tau[\ell]\hat{\mathbf{v}}_2^T \,\|\, \ldots \,\|\, \tau[2\ell]\hat{\mathbf{v}}_2^T \,\|\, \hat{\tau}^T \,\|\, \hat{\mathbf{y}}^T \,\|\, \mathfrak{m}^T \,\|\, \hat{\mathbf{s}}^T\right)^T,$$

then we will obtain the equation $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

Before going on, we define VALID as the set of $\mathbf{w} \in \{-1,0,1\}^L$ of the form:

$$\mathbf{w} = \left(\mathbf{w}_1^T \| \mathbf{w}_2^T \| g_1 \mathbf{w}_2^T \| \ldots \| g_{2\ell} \mathbf{w}_2^T \| \mathbf{g}^T \| \mathbf{w}_3^T \| \mathbf{w}_4^T \| \mathbf{w}_5^T\right)^T$$

for some $\mathbf{w}_1, \mathbf{w}_2 \in \mathsf{B}^3_{m\delta_\beta}$, $\mathbf{g} = (g_1, \ldots, g_{2\ell}) \in \mathsf{B}_{2\ell}$, $\mathbf{w}_3 \in \mathsf{B}^2_m$, $\mathbf{w}_4 \in \mathsf{CorEnc}(mN)$, and $\mathbf{w}_5 \in \mathsf{B}^3_{2m\delta_{p-1}}$. It can be checked that the constructed $\mathbf{x}$ belongs to VALID.

Step 2: *Specifying the set $\mathcal{S}$ and permutations of L elements $\{T_\pi : \pi \in \mathcal{S}\}$ for which the equivalence (11) holds.*

- Define $\mathcal{S} = \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{2\ell} \times \mathcal{S}_{2m} \times \{0,1\}^{mN} \times \mathcal{S}_{6m\delta_{p-1}}$.
- For $\pi = (\phi, \psi, \gamma, \rho, \mathbf{b}, \eta) \in \mathcal{S}$ and $\mathbf{z} = \left(\mathbf{z}_0^1 \| \mathbf{z}_0^2 \| \mathbf{z}_1 \| \ldots \| \mathbf{z}_{2\ell} \| \mathbf{g} \| \mathbf{t}_1 \| \mathbf{t}_2 \| \mathbf{t}_3\right) \in \mathbb{Z}_q^L$, where $\mathbf{z}_0^1, \mathbf{z}_0^2, \mathbf{z}_1, \ldots, \mathbf{z}_{2\ell} \in \mathbb{Z}_q^{3m\delta_\beta}$, $\mathbf{g} \in \mathbb{Z}_q^{2\ell}$, $\mathbf{t}_1 \in \mathbb{Z}_q^{2m}$, $\mathbf{t}_2 \in \mathbb{Z}_q^{2mN}$, and $\mathbf{t}_3 \in \mathbb{Z}_q^{6m\delta_{p-1}}$, we define:

$$T_\pi(\mathbf{z}) = \left(\phi(\mathbf{z}_0^1)^T \| \psi(\mathbf{z}_0^2)^T \| \psi(\mathbf{z}_{\gamma(1)})^T \| \ldots \| \psi(\mathbf{z}_{\gamma(2\ell)})^T \| \gamma(\mathbf{g})^T \| \rho(\mathbf{t}_1)^T \| E_{\mathbf{b}}(\mathbf{t}_2)^T \| \eta(\mathbf{t}_3)^T\right)^T$$

as the permutation that transforms $\mathbf{z}$ as follows:

1. It rearranges the order of the 2ℓ blocks $\mathbf{z}_1, \ldots, \mathbf{z}_{2\ell}$ according to γ.
2. It then permutes block $\mathbf{z}_0^1$ according to ϕ, blocks $\mathbf{z}_0^2$, $\{\mathbf{z}_i\}_{i=1}^{2\ell}$ according to ψ, block $\mathbf{g}$ according to γ, block $\mathbf{t}_1$ according to ρ, block $\mathbf{t}_2$ according to $E_{\mathbf{b}}$, and block $\mathbf{t}_3$ according to η.

It can be check that the equivalence (11) holds. Therefore, we can obtain a statistical ZKAoK for the given relation by running the protocol in Section 5.1.

5.5 The Underlying ZKAoK for the Group Signature Scheme

The argument system upon which our group signature scheme is built can be summarized as follows.

Common Input: Matrices $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{D}_0, \mathbf{D}_1 \in \mathbb{Z}_q^{2n \times 2m}$, $\mathbf{F} \in \mathbb{Z}_q^{4n \times 4m}$, $\mathbf{H}_{2n \times m} \in \mathbb{Z}_q^{2n \times m}$, $\mathbf{H}_{4n \times 2m} \in \mathbb{Z}_q^{4n \times 2m}$, $\mathbf{G}_0 \in \mathbb{Z}_q^{n \times 2m}$; vectors $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^m$, $\mathbf{c}_2 \in \mathbb{Z}_q^{2m}$.

Prover's Input: $\mathbf{z} \in [-\beta, \beta]^{4m}$, $\mathbf{y} \in \{0,1\}^{2m}$, $\mathbf{w} \in \{0,1\}^m$, $\mathbf{d}_1, \mathbf{d}_2 \in [-\beta, \beta]^m$, $\mathbf{s} \in [-\beta, \beta]^{2m}$, $\mathrm{id} = (\mathrm{id}[1], \ldots, \mathrm{id}[\ell])^T \in \{0,1\}^\ell$, $\mathbf{e}_0 \in [-B, B]^n$, $\mathbf{e}_1 \in [-B, B]^m$, $\mathbf{e}_2 \in [-B, B]^{2m}$.

Prover's Goal: Convince the verifier in ZK that

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

Using the same strategy as in Sections 5.3 and 5.4, we can derive a statistical ZKAoK for the above relation from the protocol given in Section 5.1. As the transformations are very similar to the ones in Section 5.4, we will only sketch main points below.

In the first step, we combine the given equations to an equation of the form:

$$\mathbf{M}\cdot\begin{pmatrix}\mathbf{d}_1\\ \mathbf{s}\\ \mathbf{z}\end{pmatrix}+\mathbf{M}_0\cdot\mathbf{d}_2+\sum_{j=1}^{\ell}\mathbf{M}_j(\mathrm{id}[j]\mathbf{d}_2)+\mathbf{M}'\cdot\begin{pmatrix}\mathbf{w}\\ \mathbf{y}\end{pmatrix}+\mathbf{M}''\cdot\begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_1\\ \mathbf{e}_2\end{pmatrix}=\mathbf{v} \bmod q,$$

where matrices $\mathbf{M}, \mathbf{M}_0, \ldots, \mathbf{M}_\ell, \mathbf{M}', \mathbf{M}''$ and vector $\mathbf{v}$ are constructed from the public input.

We then perform the techniques from Section 5.2 for the vectors $\mathbf{x}_0 = (\mathbf{d}_1^T\|\mathbf{s}^T\|\mathbf{z}^T)^T \in [-\beta,\beta]^{7m}$, $\mathbf{d}_2 \in [-\beta,\beta]^m$; $\mathbf{x}_1 = (\mathbf{w}^T\|\mathbf{y}^T)^T \in \{0,1\}^{3m}$; and $\mathbf{x}_2 = (\mathbf{e}_0^T\|\mathbf{e}_1^T\|\mathbf{e}_2^T)^T \in [-B,B]^{n+3m}$. This allows us to obtain a unified equation $\mathbf{P}\cdot\mathbf{x} = \mathbf{v} \bmod q$, and to define the sets VALID, $\mathcal{S}$, and permutations $\{T_\pi : \pi \in \mathcal{S}\}$ so that the equivalence (11) holds, in a similar manner as in Section 5.4.

Acknowledgements

We thank Damien Stehlé for useful discussions. The first author was funded by the "Programme Avenir Lyon Saint-Etienne de l'Université de Lyon" in the framework of the programme "Investissements d'Avenir" (ANR-11-IDEX-0007). San Ling, Khoa Nguyen and Huaxiong Wang were supported by the "Singapore Ministry of Education under Research Grant MOE2013-T2-1-041".

References

1. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In *Asiacrypt 2012*, number 7658, pages 4–24, 2012.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Crypto 2010*, number 6223, pages 209–236, 2010.
3. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt 2010*, volume 6110, pages 553–572, 2010.
4. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS 2009*, 2009.
5. G. Asharov, A. Jain, A. Lopez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Eurocrypt 2012*, volume 7237, pages 483–501, 2012.
6. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto 2000*, pages 255–270. Springer, 2000.
7. S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In *Asiacrypt 2015*, volume 9452 of *LNCS*, pages 3–24. Springer, 2015.
8. W. Banaszczyk. New bounds in some transference theorems in the geometry of number. *Math. Ann.*, 296:625–635, 1993.

9. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *Crypto 2009*, pages 108–125, 2009.
10. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *TCC 2008*, number 4948 in LNCS, pages 356–374. Springer, 2008.
11. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt 2003*, number 2656, pages 614–629, 2003.
12. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM-CCS 1993*, pages 62–73. ACM Press, 1993.
13. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, number 3376 in LNCS, pages 136–153. Springer, 2005.
14. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *Asiacrypt 2014*, number 8873, pages 551–572, 2014.
15. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient Zero-Knowledge Proofs for Commitments from Learning With Errors over Rings. In *ESORICS 2015*, LNCS. Springer, 2015. to appear.
16. F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, 2015.
17. D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
18. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
19. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC 2010*, volume 6056 of *LNCS*, pages 499–517. Springer, 2010.
20. X. Boyen and B. Waters. Compact group signatures without random oracles. In *Eurocrypt 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, 2006.
21. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.
22. Z. Brakerski and Y. T. Kalai. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. *IACR Cryptology ePrint Archive*, 2010:86, 2010.
23. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. In *STOC 2013*, pages 575–584. ACM, 2013.
24. E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *PKC 2000*, pages 276–292, 2000.
25. J. Camenisch and T. Gross. Efficient attributes for anonymous credentials. In *ACM-CCS 2008*, pages 345–356. ACM Press, 2008.
26. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt 2005*, number 3494 in LNCS, pages 302–321. Springer, 2005.
27. J. Camenisch, A. Kiayias, and M. Yung. On the portability of generalized schnorr proofs. In *Eurocrypt 2009*, number 5479 in LNCS. Springer, 2009.
28. J. Camenisch, S. Krenn, A. Lehmann, G.-L. Mikkelsen, G. Neven, and M.-. Pedersen. Formal treatment of privacy-enhancing credential systems. In *SAC 2015*, LNCS. Springer, 2015.

29. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Eurocrypt 2001*, number 2045 in LNCS, pages 93–118. Springer, 2001.
30. J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
31. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, number 2576 in LNCS, pages 268–289. Springer, 2002.
32. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Crypto 2004*, number 3152 in LNCS, pages 56–72. Springer, 2004.
33. J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *SCN 2012*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012.
34. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt 2010*, volume 6110, pages 523–552, 2010.
35. D. Chaum. Security without identification: Transactions ssystem to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
36. D. Chaum and E. Van Heyst. Group signatures. In *Eurocrypt 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
37. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Eurocrypt 2000*, volume 1807 of *LNCS*, pages 418–430. Springer, 2000.
38. I. Damgård, D. Hofheinz, E. Kiltz, and R. Thorbek. Public-key encryption with non-interactive opening. In *CT-RSA 2008*, volume 4964 of *LNCS*, pages 239–255. Springer, 2008.
39. C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *Vietcrypt 2006*, number 4341 in LNCS, pages 193–210. Springer, 2006.
40. M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, and H. Wang. A provably secure group signature scheme from code-based assumptions. In *Asiacrypt 2015*, LNCS. Springer, 2015. `http://eprint.iacr.org/`.
41. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto 1986*, pages 186–194. Springer, 1987.
42. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 2009*, pages 169–178, 2009.
43. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206, 2008.
44. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 1985*, pages 291–304. ACM, 1985.
45. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC 2013*, pages 545–554. ACM Press, 2013.
46. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from lwe. In *Crypto 2015*, number 9216, pages 503–523, 2015.
47. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Asiacrypt 2010*, volume 2647 of *LNCS*, pages 395–412. Springer, 2010.
48. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In *Asiacrypt 2008*, number 5350 in LNCS, pages 179–197. Springer, 2008.
49. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
50. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.

51. S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *Crypto 2009*, volume 5677 of *LNCS*, pages 654–670. Springer, 2009.
52. A. Jain, S. Krenn, K. Pietrzak, and A. Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *Asiacrypt 2012*, volume 7658 of *LNCS*, pages 663–680. Springer, 2012.
53. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Asiacrypt 2008*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.
54. A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. In *Asiacrypt 2007*, number 4833 in LNCS, pages 181–199. Springer, 2007.
55. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt 2005*, number 3494 in LNCS, pages 198–214. Springer, 2005.
56. A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *Int. Journal of Security and Networks*, 1(1):24–45, 2006.
57. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *Asiacrypt 2013*, volume 8270 of *LNCS*, pages 41–61. Springer, 2013.
58. A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *PKC 2014*, volume 8383 of *LNCS*, pages 345–361. Springer, 2014.
59. A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *Eurocrypt 2014*, volume 8441 of *LNCS*, pages 239–256. Springer, 2014.
60. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *Eurocrypt 2016*, LNCS. Springer, 2016. To appear.
61. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the isis problem, and applications. In *PKC 2013*, volume 7778, pages 107–124. Springer, 2013.
62. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 427–449. Springer, 2015.
63. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *Public Key Cryptography*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
64. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Eurocrypt 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
65. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
66. P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC 2015*, volume 9020 of *LNCS*, pages 401–426. Springer, 2015.
67. C. Papamanthou, E. Shi, R. Tamassia, and K. Yi. Streaming Authenticated Data Structures. In *EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 353–370. Springer, 2013.
68. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC 2009*, pages 333–342, 2009.
69. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93, 2005.
70. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Asiacrypt 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.

71. Y. Sakai, J. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *PKC 2012*, volume 7293 of *LNCS*, pages 715–732. Springer, springer edition, 2012.
72. J. Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6):1757–1768, 1996.
73. X. Xie, R. Xue, and M. Wang. Zero knowledge proofs from ring-lwe. In *CANS 2013*, volume 8257 of *LNCS*, page 5773. Springer, 2013.

A Definitions for Dynamic Group Signatures

This section recalls the syntax and the security definitions of dynamic group signatures based on the model of Kiayias and Yung [56].

A *group signature* allows a group member to attest that a message was provided by a member of a *group* without being altered during the process and preserving the *anonymity* of the users. This primitive was introduced by Bellare, Micciancio and Warinschi [11] in 2003 and was extended to dynamic groups by Bellare, Shi and Zhang (BSZ) in 2005 [13].

In the setting of *dynamic groups*, the syntax of group signatures includes an interactive protocol which allows users to register as new members of the group at any time. The syntax and the security model are those defined by Kiayias and Yung [56]. Like the very similar BSZ model [13], the Kiayias-Yung (KY) model assumes an interactive *join* protocol whereby a prospective user becomes a group member by interacting with the group manager. This protocol provides the user with a membership certificate, cert_i, and a membership secret, sec_i.

We denote by $N \in \mathsf{poly}(n)$ the maximal number of group members.

Definition 4 (Dynamic Group Signature). *A* dynamic group signature *scheme consists of the following algorithms or protocols.*

Setup$(1^n, N)$**:** *given a security parameter n and a maximal number of group members $N \in \mathbb{N}$, this algorithm is run by a trusted party to generate a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Each key is given to the appropriate authority while $\mathcal{Y}$ is made public. The algorithm also initializes a public state St comprising a set data structure $St_{\mathsf{users}} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$.*
In the following, all algorithms have access to the public parameters $\mathcal{Y}$.

Join: *is an* interactive *protocol between the group manager GM and a user $\mathcal{U}_i$ where the latter becomes a group member. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take $\mathcal{Y}$ as input. The execution, denoted as $[\mathsf{J}_{\mathsf{user}}(n, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(n, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$, ends with user $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol is successful, the group manager updates the public state St by setting $St_{\mathsf{users}} := St_{\mathsf{users}} \cup \{i\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle i, \mathsf{transcript}_i \rangle$.*

Sign($\mathsf{cert}_i, \mathsf{sec}_i, M$**):** *given a membership certificate cert_i, a membership secret sec_i and a message M, this* probabilistic *algorithm outputs a signature σ.*

Verify(σ, M**):** *given a signature σ, a message M and a group public key $\mathcal{Y}$, this* deterministic *algorithm returns either* 0 *or* 1.

Open($\mathcal{S}_{\mathsf{OA}}, M, \sigma$**):** *takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state St. It outputs $i \in St_{\mathsf{users}} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.*

Each membership certificate contains a unique tag *that identifies the user.*

The correctness requirement basically captures that, if all parties *honestly* run the protocols, all algorithms are correct with respect to their specification described as above.

The Kiayias-Yung model [56] considers three security notions: the security against *misidentification attacks* requires that, even if the adversary can introduce users under its control in the group, it cannot produce a signature that traces outside the set of dishonest users. The notion of security against *framing attacks* implies that honest users can never be accused of having signed messages that they did not sign, even if the whole system conspired against them. And finally the *anonymity* property is also formalized by granting the adversary access to a signature opening oracle as in the models of [13].

Correctness for Dynamic Group Signatures. Following the Kiayias-Yung terminology [56], we say that a public state St is *valid* if it can be reached from $St = (\emptyset, \varepsilon)$ by a Turing machine having oracle access to J_{GM}. Also, a state St' is said to *extend* another state St if it is within reach from St.

Moreover, as in [56], when we write $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, it means that there exists coin tosses ϖ for J_{GM} and J_{user} such that, for some valid public state St', the execution of the interactive protocol $[\mathsf{J}_{\mathsf{user}}(n, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(n, St', \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})](\varpi)$ provides $\mathsf{J}_{\mathsf{user}}$ with $\langle i, \mathsf{sec}_i, \mathsf{cert}_i \rangle$.

Definition 5 (Correctness). *A dynamic group signature scheme is correct if the following conditions are all satisfied:*

(1) In a valid state St, $|St_{users}| = |St_{trans}|$ always holds and two distinct entries of St_{trans} always contain certificates with distinct tag.

(2) If $[\mathsf{J}_{\mathsf{user}}(n, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(n, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$ is run by two honest parties following the protocol and at the end $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ is obtained by $\mathsf{J}_{\mathsf{user}}$, then it holds that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.

(3) For each $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ such that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, satisfying condition 2, it always holds that:

$$\mathsf{Verify}\big(\mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M), M, \mathcal{Y}\big) = 1$$

(4) For any outcome $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ of the interaction $[\mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., St, ., .)]$ for some valid state St, if $\sigma = \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$, then

$$\mathsf{Open}(M, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') = i.$$

We formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes the various oracles available in the attack games. It is initialized as $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(n, N)$. It includes the (initially empty) set St_{users} of group members and a dynamically growing database St_{trans} storing the transcripts of previously executed join protocols.
- $n = |St_{users}| < N$ denotes the current cardinality of the group.

- Sigs: is a database of signatures created by the signing oracle. Each entry consists of a triple (i, M, σ) indicating that message M was signed by user i.
- U^a: is the set of users that were introduced by the adversary in the system in an execution of the join protocol.
- U^b: is the set of honest users that the adversary, acting as a dishonest group manager, introduced in the system. For these users, the adversary obtains the transcript of the join protocol but not the user's membership secret.

When mounting attacks, adversaries will be granted access to the following oracles:

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively.
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under his control in the group. On behalf of the GM, the interface runs J_{GM} in interaction with the $\mathsf{J}_{\mathsf{user}}$-executing adversary who plays the role of the prospective user in the join protocol. If this protocol successfully ends, the interface increments n, updates St by inserting the new user n in both sets St_{users} and U^a. It also sets $St_{\mathsf{trans}} := St_{\mathsf{trans}}||\langle n, \mathsf{transcript}_n\rangle$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a corrupted group manager, to introduce new honest group members of his/her choice. The interface triggers an execution of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the adversary who runs J_{GM}. If the protocol successfully completes, the interface increments n, adds user n to St_{users} and U^b and sets $St_{\mathsf{trans}} := St_{\mathsf{trans}}||\langle n, \mathsf{transcript}_n\rangle$. It stores the membership certificate cert_n and the membership secret sec_n in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a message M, an index i, the interface checks whether the private area of $\mathsf{state}_{\mathcal{I}}$ contains a certificate cert_i and a membership secret sec_i. If no such elements $(\mathsf{cert}_i, \mathsf{sec}_i)$ exist or if $i \not\in U^b$, the interface returns $\perp$. Otherwise, it outputs a signature σ on behalf of user i and also sets $\mathsf{Sigs} \leftarrow \mathsf{Sigs}||(i, M, \sigma)$.
- Q_{open}: when this oracle is invoked on input of a valid pair (M, σ), the interface runs algorithm Open using the current state St. When S is a set of pairs of the form (M, σ), $Q_{\mathsf{open}}^{\neg S}$ denotes a restricted oracle that only applies the opening algorithm to pairs (M, σ) which are not in S.
- Q_{read} and Q_{write}: are used by the adversary to read and write the content of $\mathsf{state}_{\mathcal{I}}$. Namely, at each invocation, Q_{read} outputs the whole $\mathsf{state}_{\mathcal{I}}$ but the public/private keys and the private part of $\mathsf{state}_{\mathcal{I}}$ where membership secrets are stored after $Q_{\mathsf{b\text{-}join}}$-queries. By using Q_{write}, the adversary can modify $\mathsf{state}_{\mathcal{I}}$ at will as long as it does not remove or alter elements of St_{users}, St_{trans} or invalidate the public state St: for example, the adversary is allowed to create dummy users as long as he/she does not re-use already existing certificate tags.

Based on the above syntax, the announced security properties are formalized as follows.

Security Against Misidentification Attacks. In a misidentification attack, the adversary can corrupt the opening authority using the Q_{keyOA} oracle. Moreover, he/she can also introduce malicious users in the group via $Q_{\mathsf{a\text{-}join}}$-queries. His/her purpose is to come up with a valid signature $\sigma^\star$. He/she succeeds if the produced signature $\sigma^\star$ does not open to any adversarially-controlled.

Definition 6. *A dynamic group signature scheme is secure against* misidentification attacks *if, for any* PPT *adversary $\mathcal{A}$ involved in the experiment hereunder,the advantage of function of the adversary $\mathcal{A}$ satisfies:*

$$\mathbf{Adv}_{\mathcal{A}}^{\text{mis-id}}(n) = \Pr\left[\mathbf{Exp}_{\mathcal{A}}^{\text{mis-id}}(n) = 1\right] \in \mathsf{negl}\,(n)$$

Experiment 1: Experiment $\mathbf{Exp}_{\mathcal{A}}^{mis\text{-}id}(n)$

```
$\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(n, N)$;
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{read}}, Q_{\mathsf{keyOA}})$;
if $\mathsf{Verify}(\sigma^\star, M^\star, \mathcal{Y}) = 0$ then
    return $0$;
$i = \mathsf{Open}(M^\star, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
if $i \notin U^a$ then
    return $1$;
return $0$;
```

Non-Frameability. Framing attacks consider the situation where the entire system, including the group manager and the opening authority, is colluding against some honest user. The adversary can corrupt the group manager as well as the opening authority (via oracles Q_{keyGM} and Q_{keyOA}, respectively). He/she is also allowed to introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users sign messages and create dummy users using Q_{write}. The adversary eventually aims at framing an honest group member.

Definition 7. *A dynamic group signature scheme is secure against* framing attacks *if, for any* PPT *adversary $\mathcal{A}$ involved in the experiment below, it holds that* $\mathbf{Adv}_{\mathcal{A}}^{\text{fra}}(n) = \Pr\left[\mathbf{Exp}_{\mathcal{A}}^{\text{fra}}(n) = 1\right] \in \mathsf{negl}\,(n)$.

Full Anonymity. The notion of anonymity is formalized by means of a game involving a two-stage adversary. The first stage is called play stage and allows the adversary $\mathcal{A}$ to modify $\mathsf{state}_{\mathcal{I}}$ via Q_{write}-queries and open arbitrary signatures by probing Q_{open}. When the play stage ends, $\mathcal{A}$ chooses a message $M^\star$ as well as two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$ and $(\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, consisting of a valid membership certificate and a corresponding membership secret. Then, the challenger flips a coin $d \leftarrow \{0, 1\}$

Experiment 2: Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{fra}}(n)$

```
$\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(n, N)$;
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
if $\mathsf{Verify}(\sigma^\star, M^\star, \mathcal{Y}) = 0$ then
    return $0$;
if $i = \mathsf{Open}(M^\star, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') \notin U^b$ then
    return $0$;
if $\big(\bigwedge_{j \in U^b \ s.t. \ j=i} (j, M^\star, *) \notin \mathsf{Sigs}\big)$ then
    return $1$;
return $0$;
```

and computes a challenge signature $\sigma^\star$ using $(\mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$. The adversary is given $\sigma^\star$ with the task of eventually guessing the bit $d \in \{0,1\}$. Before doing so, he/she is allowed further oracle queries throughout the second stage, called guess stage, but is restricted not to query Q_{open} for $(M^\star, \sigma^\star)$.

Definition 8. *A dynamic group signature scheme is fully anonymous if, for any PPT adversary $\mathcal{A}$ involved in the following experiment below, we have $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{anon}}(n) := |\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{anon}}(n) = 1] - 1/2| \in \mathsf{negl}(n)$.*

Experiment 3: Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{anon}}(n)$

```
$\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(n)$;
$\big(aux, M^\star, (\mathsf{sec}_0^\star, \mathsf{cert}_0^\star), (\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)\big) \leftarrow \mathcal{A}(\mathsf{play};\ Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
if $\neg(\mathsf{cert}_b^\star \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b^\star)$ for $b \in \{0,1\}$ then
    return $0$;
if $\mathsf{cert}_0^\star = \mathsf{cert}_1^\star$ then
    return $0$;
Picks random $d \leftarrow \{0,1\}$; $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_d^\star, \mathsf{sec}_d^\star, M^\star)$;
$d' \leftarrow \mathcal{A}(\mathsf{guess};\ \sigma^\star, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}^{\neg\{(M^\star, \sigma^\star)\}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
if $d' = d$ then
    return $1$;
return $0$;
```

B Deferred Proofs for the Signature with Efficient Protocols

In the security proof of the signature with efficient protocols, we make use of the Rényi divergence in a similar way to [7] in the proof of Theorem 2.

B.1 The Rényi Divergence

Instead of the classical statistical distance we sometimes use the Rényi divergence, which is a measurement of the distance between two distributions. Its use in security proofs for lattice-based systems was first considered by Bai *et al.* [7].

Definition 9 (Rényi divergence). *For any two discrete distributions P and Q such that* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$*, and $a \in]1, +\infty[$, we define the* Rényi divergence *of order a by:*

$$R_a(P||Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}$$

We define the Rényi divergences of orders 1 and $+\infty$ by:

$$R_1(P||Q) = \exp\left(\sum_{x \in \mathrm{Supp}(P)} P(x) \log \frac{P(x)}{Q(x)} \right) \text{ and } R_\infty(P||Q) = \max_{x \in \mathrm{Supp}(P)} \frac{P(x)}{Q(x)}.$$

The divergence R_1 is the (exponential) of the Kullback-Leibler divergence.

We will focus on the following properties of the Rényi divergence, the proofs can be found in [59].

Lemma 9 ([7, Le. 2.7]). *Let $a \in [1, +\infty]$. Let P and Q denote distributions with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$*. Then the following properties hold:*

Log. Positivity: $R_a(P||Q) \geq R_a(P||P) = 1$

Data Processing Inequality: $R_a(P^f||Q^f) \leq R_a(P||Q)$ *for any function f, where P^f denotes the distribution of $f(y)$ induced by sampling $y \hookleftarrow P$ (resp. $y \hookleftarrow Q$)*

Multiplicativity: *Assume P and Q are two distributions of a pair of random variables (Y_1, Y_2). For $i \in \{1, 2\}$, let P_i (resp. Q_i) denote the marginal distribution of Y_i under P (resp. Q), and let $P_{2|1}(\cdot|y_1)$ (resp. $Q_{2|1}(\cdot|y_1)$) denote the conditional distribution of Y_2 given that $Y_1 = y_1$. Then we have:*
- $R_a(P||Q) = P_a(P_1||Q_1) \cdot R_a(P_2||Q_2)$ *if Y_B and Y_2 are independent;*
- $R_a(P||Q) \leq R_\infty(P_1||Q_1) \cdot max_{y_1 \in X} R_a\left(P_{2|1}(\cdot|y_1)||Q_{2|1}(\cdot|y_1)\right)$.

Probability Preservation: *Let $A \subseteq$* $\mathrm{Supp}(Q)$ *be an arbitrary event. If $a \in]1, +\infty[$, then $Q(A) \geq P(A)^{\frac{a}{a-1}} / R_a(P||Q)$. Further we have:*

$$Q(A) \geq P(A)/R_\infty(P||Q)$$

Weak Triangle Inequality: *Let P_1, P_2, P_3 be three distributions with* $\mathrm{Supp}(P_1) \subseteq \mathrm{Supp}(P_2) \subseteq \mathrm{Supp}(P_3)$*. Then we have:*

$$R_a(P_1||P_3) \leq \begin{cases} R_a(P_1||P_2) \cdot R_\infty(P_2||P_3), \\ R_\infty(P_1||P_2)^{\frac{a}{a-1}} \cdot R_a(P_2||P_3) & \text{if } a \in]1, +\infty[. \end{cases}$$

In our proofs, we mainly use the probability preservation to bound the probabilities during hybrid games where the two distribution are not statistically indistinguishable in the attacker point of view.

B.2 Proof of Lemma 6

Proof. Let $\mathcal{A}$ be a PPT adversary that can mount a Type I attack with non-negligible success probability ε. We construct a PPT algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the $\mathsf{SIS}_{n,m,q,\beta'}$ assumption. It takes as input $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ and computes $\mathbf{v} \in \Lambda_q^{\perp}(\bar{\mathbf{A}})$ with $0 < \|\mathbf{v}\| \leq \beta'$.

Algorithm $\mathcal{B}$ first chooses the ℓ-bit strings $\tau^{(1)}, \ldots, \tau^{(Q)} \hookleftarrow U(\{0,1\}^{\ell})$ to be used in signing queries. As in [51], it guesses the shortest prefix such that the string $\tau^{\star}$ contained in $\mathcal{A}$'s forgery differs from all prefixes of $\tau^{(1)}, \ldots, \tau^{(Q)}$. To this end, $\mathcal{B}$ chooses $i^{\dagger} \hookleftarrow U(\{1, \ldots, Q\})$ and $t^{\dagger} \hookleftarrow U(\{1, \ldots, \ell\})$ so that, with probability $1/(Q \cdot \ell)$, the longest common prefix between $\tau^{\star}$ and one of the $\{\tau^{(i)}\}_{i=1}^{Q}$ is the string $\tau^{\star}[1] \ldots \tau^{\star}[t^{\dagger}-1] = \tau^{(i^{\dagger})}[1] \ldots \tau^{(i^{\dagger})}[t^{\dagger}-1] \in \{0,1\}^{t^{\dagger}-1}$ comprised of the first $(t^{\dagger}-1)$-th bits of $\tau^{\star} \in \{0,1\}^{\ell}$. We define $\tau^{\dagger} \in \{0,1\}^{t^{\dagger}}$ as the $t^{\dagger}$-bit string $\tau^{\dagger} = \tau^{\star}[1] \ldots \tau^{\star}[t^{\dagger}]$. By construction, with probability $1/(Q \cdot \ell)$, we have $\tau^{\dagger} \not\in \{\tau_{|t^{\dagger}}^{(1)}, \ldots, \tau_{|t^{\dagger}}^{(Q)}\}$, where $\tau_{|t^{\dagger}}^{(i)}$ denotes the $t^{\dagger}$-th prefix of $\tau^{(i)}$ for each $i \in \{1, \ldots, Q\}$.

Then, $\mathcal{B}$ runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{T_C}$ of $\Lambda_q^{\perp}(\mathbf{C})$ with $\|\widetilde{\mathbf{T_C}}\| \leq \mathcal{O}(\sqrt{n \log q})$. Then, it picks $\ell + 1$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_{\ell} \in \mathbb{Z}^{m \times m}$, where each matrix $\mathbf{Q}_i$ has its columns sampled independently from $D_{\mathbb{Z}^m,\sigma}$. The reduction $\mathcal{B}$ defines the matrices $\{\mathbf{A}_j\}_{j=0}^{\ell}$ as

$$\begin{cases} \mathbf{A}_0 = \bar{\mathbf{A}} \cdot \mathbf{Q}_0 + (\sum_{j=1}^{t^{\dagger}} \tau^{\star}[j]) \cdot \mathbf{C} & \\ \mathbf{A}_j = \bar{\mathbf{A}} \cdot \mathbf{Q}_j + (-1)^{\tau^{\star}[j]} \cdot \mathbf{C}, & \text{for } j \in [j, t^{\dagger}] \\ \mathbf{A}_j = \bar{\mathbf{A}} \cdot \mathbf{Q}_j, & \text{for } j \in [t^{\dagger}+1, \ell] \end{cases}$$

It also sets $\mathbf{A} = \bar{\mathbf{A}}$. We note that we have

$$\begin{aligned} \mathbf{A}_{\tau^{(i)}} &= \Big[\bar{\mathbf{A}} \Big| \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i)}[j] \mathbf{A}_j \Big] \\ &= \Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i)}[j] \mathbf{Q}_j) + (\sum_{j=1}^{t^{\dagger}} \tau^{\star}[j] + (-1)^{\tau^{\star}[j]} \tau^{(i)}[j]) \cdot \mathbf{C} \Big] \\ &= \Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i)}[j] \mathbf{Q}_j) + h_{\tau^{(i)}} \cdot \mathbf{C} \Big] \end{aligned}$$

where $h_{\tau^{(i)}} \in [1, t^{\dagger}] \subset [1, \ell]$ stands for the Hamming distance between $\tau_{|t^{\dagger}}^{(i)}$ and $\tau_{|t^{\dagger}}^{\dagger}$. Note that, with probability $1/(Q \cdot \ell)$ and since $q > \ell$, we have $h_{\tau^{(i)}} \neq 0 \bmod q$ whenever $\tau_{|t^{\dagger}}^{(i)} \neq \tau_{|t^{\dagger}}^{\star}$.

Next, $\mathcal{B}$ picks a random short matrix $\mathbf{R} \hookleftarrow \mathbb{Z}^{m \times m}$ which has its columns independently sampled from $D_{\mathbb{Z}^m,\sigma}$ and computes

$$\mathbf{D} = \bar{\mathbf{A}} \cdot \mathbf{R}.$$

Finally, $\mathcal{B}$ samples short vectors $\mathbf{e}_u \in D_{\mathbb{Z}^m,\sigma_1}$ and computes the vector $\mathbf{u} \in \mathbb{Z}_q^n$ as $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u \in \mathbb{Z}_q^n$. The public key

$$PK := \big(\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \{\mathbf{D}_k\}_{k=0}^{N},\ \mathbf{D},\ \mathbf{u}\big)$$

is given to $\mathcal{A}$.

At the i-th signing query $\mathsf{Msg}^{(i)} = (\mathfrak{m}_1^{(i)}, \ldots, \mathfrak{m}_N^{(i)}) \in (\{0,1\}^m)^N$, $\mathcal{B}$ can use the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ to generate a signature. To do this, $\mathcal{B}$ first samples $\boldsymbol{s}^{(i)} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_1}$ and computes a vector $\mathbf{u}_M \in \mathbb{Z}_q^m$ as

$$\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i)} + \mathbf{D}_0 \cdot \mathbf{s}^{(i)}\big) \mod q.$$

Using $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$, $\mathcal{B}$ can then sample a short vector $\mathbf{v}^{(i)} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_{\tau^{(i)}}),\sigma}$ such that $(\tau^{(i)}, \mathbf{v}^{(i)}, \mathbf{s}^{(i)})$ satisfies the verification equation (2).

When $\mathcal{A}$ halts, it outputs a valid signature $sig^\star = (\tau^{(i^\dagger)}, \mathbf{v}^\star, \mathbf{s}^\star)$ on a message $\mathsf{Msg}^\star = (\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$ with $\|\mathbf{v}^\star\| \leq \sigma\sqrt{2m}$ and $\|\mathbf{s}^\star\| \leq \sigma_1\sqrt{2m}$. At this point, $\mathcal{B}$ aborts and declares failure if it was unfortunate in its choice of $i^\dagger \in \{1, \ldots, Q\}$ and $t^\dagger \in \{1, \ldots, \ell\}$. Otherwise, with probability $1/(Q \cdot \ell)$, $\mathcal{B}$ correctly guessed $i^\dagger \in \{1, \ldots, Q\}$ and $t^\dagger \in \{1, \ldots, \ell\}$, in which case it can solve the given SIS instance as follows.

If we parse $\mathbf{v}^\star \in \mathbb{Z}^{2m}$ as $(\mathbf{v}_1^{\star T} \mid \mathbf{v}_2^{\star T})^T$ with $\mathbf{v}_1^\star, \mathbf{v}_2^\star \in \mathbb{Z}^m$, we have the equality

$$\begin{aligned}
\Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \textstyle\sum_{j=1}^{\ell} \tau^\star[j]\mathbf{Q}_j)\Big] \;\cdot\; \begin{bmatrix} \mathbf{v}_1^\star \\ \hline \mathbf{v}_2^\star \end{bmatrix}& \\
&= \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star\big) \mod q \\
&= \bar{\mathbf{A}} \cdot \Big(\mathbf{e}_u + \mathbf{R} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star\big)\Big) \bmod q,
\end{aligned}$$

which implies that the vector

$$\mathbf{w} = \mathbf{v}_1^\star + (\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^\star[j]\mathbf{Q}_j) \cdot \mathbf{v}_2^\star - \mathbf{e}_u - \mathbf{R} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star\big) \in \mathbb{Z}^m$$

is in $\Lambda^\perp(\bar{\mathbf{A}})$. Moreover, with overwhelming probability, this vector is non-zero since, in $\mathcal{A}$'s view, the distribution of $\mathbf{e}_u \in \mathbb{Z}^m$ is $D_{\Lambda_q^{\mathbf{u}}(\bar{\mathbf{A}}),\sigma_1}$, which ensures that $\mathbf{e}_u$ is statistically hidden by the syndrome $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u$. Finally, the norm of $\mathbf{w}$ is smaller than $\beta' = m^{3/2}\sigma^2(\ell + 3) + m^{1/2}\sigma_1$ which yields a valid solution of the given $\mathsf{SIS}_{n,m,q,\beta'}$ instance with overwhelming probability. □

B.3 Proof of Lemma 7

Proof. We prove the result using a sequence of games. For each i, we denote by W_i the event that the adversary wins by outputting a Type II forgery in Game i.

Game 0**:** This is the real game where, at the i-th signing query $\mathsf{Msg}^{(i)} = (\mathfrak{m}_1^{(i)}, \ldots, \mathfrak{m}_N^{(i)})$, the adversary obtains a signature $sig^{(i)} = (\tau^{(i)}, \mathbf{v}^{(i)}, \mathbf{s}^{(i)})$ for each $i \in \{1,\ldots,Q\}$ from the signing oracle. At the end of the game, the adversary outputs a forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$ on a message $\mathsf{Msg}^\star = (\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$. By hypothesis, the adversary's advantage is $\varepsilon = \Pr[W_0]$. We assume w.l.o.g. that the random ℓ-bit strings $\tau^{(1)}, \ldots, \tau^{(Q)}$ are chosen at the very beginning of the game. Since $(\mathsf{Msg}^\star, sig^\star)$ is a Type II forgery, there exists an index $i^\star \in \{1,\ldots,Q\}$ such that $\tau^\star = \tau^{(i^\star)}$.

Game 1: This game is identical to Game 0 with the difference that the reduction aborts the experiment in the unlikely event that, in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, $\tau^\star$ coincides with more than one of the random ℓ-bit strings $\tau^{(1)}, \ldots, \tau^{(Q)}$ used by the challenger. If we call F_1 the latter event, we have $\Pr[F_1] < Q^2/2^\ell$ since we are guaranteed to have $\neg F_1$ as long as no two $\tau^{(i)}$, $\tau^{(i')}$ collide. Given that Game 1 is identical to Game 0 until F_1 occurs, we have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] < Q^2/2^\ell$.

Game 2: This game is like Game 1 with the following difference. At the outset of the game, the challenger $\mathcal{B}$ chooses a random index $i^\dagger \hookleftarrow U(\{1,\ldots,Q\})$ as a guess that $\mathcal{A}$'s forgery will recycle the ℓ-bit string $\tau^{(i^\dagger)} \in \{0,1\}^\ell$ of the $i^\dagger$-th signing query. When $\mathcal{A}$ outputs its Type II forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, the challenger aborts in the event that $\tau^{(i^\dagger)} \neq \tau^\star$ (i.e., $i^\dagger \neq i^\star$). Since the choice of $i^\dagger$ in $\{1,\ldots,Q\}$ is independent of $\mathcal{A}$'s view, we have $\Pr[W_2] = \Pr[W_1]/Q$.

Game 3: In this game, we modify the key generation phase and the way to answer signing queries. First, the challenger $\mathcal{B}$ randomly picks $h_0, h_1, \ldots, h_\ell \in \mathbb{Z}_q$ subject to the constraints

$$\begin{aligned} h_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot h_j &= 0 \bmod q \\ h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot h_j &\neq 0 \bmod q \qquad\qquad i \in \{1,\ldots,Q\} \setminus \{i^\dagger\} \end{aligned}$$

It runs $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, $(\mathbf{D}_0, \mathbf{T}_{\mathbf{D}_0}) \leftarrow \mathsf{TrapGen}(1^{2n}, 1^{2m}, q)$ so as to obtain statistically random matrices $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{D}_0 \in \mathbb{Z}_q^{2n \times 2m}$ with trapdoors $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$, $\mathbf{T}_{\mathbf{D}_0} \in \mathbb{Z}^{2m \times 2m}$ consisting of short bases of $\Lambda_q^\perp(\mathbf{C})$ and $\Lambda_q^\perp(\mathbf{D}_0)$, respectively. Then, $\mathcal{B}$ chooses a uniformly random $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ and re-randomizes it using short matrices $\mathbf{S}, \mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell \hookleftarrow \mathbb{Z}^{m \times m}$, which are obtained by sampling their columns from the distribution $D_{\mathbb{Z}^m,\sigma}$. Namely, from $\mathbf{D} \in \mathbb{Z}_q^{n \times m}$, $\mathcal{B}$ defines

$$\begin{aligned} \mathbf{A} &= \mathbf{D} \cdot \mathbf{S} \\ \mathbf{A}_0 &= \mathbf{D} \cdot \mathbf{S}_0 + h_0 \cdot \mathbf{C} \\ \mathbf{A}_j &= \mathbf{D} \cdot \mathbf{S}_j + h_j \cdot \mathbf{C} \qquad\qquad \forall j \in \{1,\ldots,\ell\} \end{aligned} \tag{17}$$

In addition, $\mathcal{B}$ picks random matrices $\mathbf{D}_1, \ldots, \mathbf{D}_N \hookleftarrow U(\mathbb{Z}_q^{2n \times 2m})$ and a random vector $\mathbf{c}_M \hookleftarrow U(\mathbb{Z}_q^{2n})$. It samples short vectors $\mathbf{v}_1, \mathbf{v}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and computes

$\mathbf{u} \in \mathbb{Z}_q^n$ as $\mathbf{u} = \mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} - \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M) \bmod q$, where

$$\begin{aligned} \mathbf{A}_{\tau^{(i^\dagger)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{A}_j \Big] \\ &= \Big[\mathbf{D} \cdot \mathbf{S} \;\Big|\; \mathbf{D} \cdot (\mathbf{S}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j) \Big] . \end{aligned}$$

The adversary's signing queries are then answered as follows.

- At the i-th signing query $(\mathfrak{m}_1^{(i)}, \ldots, \mathfrak{m}_N^{(i)})$, whenever $i \neq i^\dagger$, we have

$$\begin{aligned} \mathbf{A}_{\tau^{(i)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{A}_j \Big] \\ &= \Big[\mathbf{A} \;\Big|\; \mathbf{D} \cdot (\mathbf{S}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{S}_j) + h_{\tau^{(i)}} \cdot \mathbf{C} \Big] \in \mathbb{Z}_q^{n \times 2m}, \end{aligned}$$

  with $h_{\tau^{(i)}} = h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot h_j \neq 0$. This implies that $\mathcal{B}$ can use the trapdoor $\mathbf{T}_{\mathbf{C}} \in \mathbb{Z}^{m \times m}$ to generate a signature. To this end, $\mathcal{B}$ first samples a discrete Gaussian vector $\boldsymbol{s}^{(i)} \hookleftarrow D_{\mathbb{Z}^{2m}, \sigma_1}$ and computes $\mathbf{u}_M \in \mathbb{Z}_q^n$ as

$$\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i)} + \mathbf{D}_0 \cdot \mathbf{s}^{(i)}) \mod q.$$

  Then, using $\mathbf{T}_{\mathbf{C}} \in \mathbb{Z}^{m \times m}$, it samples a short vector $\mathbf{v}^{(i)} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_{\tau^{(i)}}), \sigma}$ such that $(\tau^{(i)}, \mathbf{v}^{(i)}, \mathbf{s}^{(i)})$ satisfies (2).
- At the $i^\dagger$-th signing query $(\mathfrak{m}_1^{(i^\dagger)}, \ldots, \mathfrak{m}_N^{(i^\dagger)})$, we have

$$\begin{aligned} \mathbf{A}_{\tau^{(i^\dagger)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{A}_j \Big] \\ &= \Big[\mathbf{D} \cdot \mathbf{S} \;\Big|\; \mathbf{D} \cdot (\mathbf{S}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j) \Big] \in \mathbb{Z}_q^{n \times 2m} \end{aligned} \tag{18}$$

  due to the constraint $h_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot h_j = 0 \bmod q$. To answer the query, $\mathcal{B}$ uses the trapdoor $\mathbf{T}_{\mathbf{D}_0} \in \mathbb{Z}^{2m \times 2m}$ of $\Lambda_q^\perp(\mathbf{D}_0)$ to sample a short vector $\mathbf{s}^{(i^\dagger)} \in D_{\Lambda_q^{\mathbf{c}'_M}(\mathbf{D}_0), \sigma_1}$, where $\mathbf{c}'_M = \mathbf{c}_M - \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} \in \mathbb{Z}_q^{2n}$. The obtained vector $\mathbf{s}^{(i^\dagger)} \in \mathbb{Z}^{2m}$ thus verifies

$$\mathbf{D}_0 \cdot \mathbf{s}^{(i^\dagger)} = \mathbf{c}_M - \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} \mod q, \tag{19}$$

  and $\mathcal{A}$ receives $sig^{(i^\dagger)} = (\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}^{(i^\dagger)})$, where $\mathbf{v}^{(i^\dagger)} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T$. By construction, the returned signature $sig^{(i^\dagger)}$ satisfies

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathbf{s}^{(i^\dagger)} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)}\big) \mod q,$$

  and the distribution of $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}^{(i^\dagger)})$ is statistically the same as in Game 2.

We conclude that $\Pr[W_2]$ is negligibly far apart from $\Pr[W_3]$ since, by the Leftover Hash Lemma (see [3, Lemma 13]), the public key PK in $\mathsf{Game}\ 3$ is statistically close to its distribution in $\mathsf{Game}\ 2$.

In $\mathsf{Game}\ 3$, we claim that the challenger $\mathcal{B}$ can use $\mathcal{A}$ to solve the SIS problem by finding a short vector of $\Lambda_q^{\perp}(\mathbf{D})$ with probability $\Pr[W_3]$. Indeed, with probability $\Pr[W_3]$, the adversary outputs a valid signature $sig^\star = (\tau^{(i^\dagger)}, \mathbf{v}^\star, \mathbf{s}^\star)$ on a message $\mathsf{Msg}^\star = (\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$ with $\|\mathbf{v}^\star\| \leq \sigma\sqrt{2m}$ and $\|\mathbf{s}^\star\| \leq \sigma_1\sqrt{2m}$. If we parse $\mathbf{v}^\star \in \mathbb{Z}^{2m}$ as $(\mathbf{v}_1^{\star T} \mid \mathbf{v}_2^{\star T})^T$ with $\mathbf{v}_1^\star, \mathbf{v}_2^\star \in \mathbb{Z}^m$, we have the equality

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\frac{\mathbf{v}_1^\star}{\mathbf{v}_2^\star}\right] = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star) \mod q. \tag{20}$$

Due to the way $\mathbf{u} \in \mathbb{Z}_q^n$ was defined at the outset of the game, $\mathcal{B}$ also knows short vectors $\mathbf{v}^{(i^\dagger)} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ such that

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\frac{\mathbf{v}_1}{\mathbf{v}_2}\right] = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M) \bmod q. \tag{21}$$

Relation (19) implies that $\mathbf{c}_M \neq \mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star \bmod q$ by hypothesis. It follows that $\mathsf{bin}(\mathbf{c}_M) - \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star)$ is a non-zero vector in $\{-1, 0, 1\}^m$. Subtracting (21) from (20), we get

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\frac{\mathbf{v}_1^\star - \mathbf{v}_1}{\mathbf{v}_2^\star - \mathbf{v}_1}\right] = \mathbf{D} \cdot \big(\mathsf{bin}(\mathbf{c}_M) - \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star)\big) \mod q,$$

which implies

$$\begin{aligned} \Big[\mathbf{D} \cdot \mathbf{S} \;\Big|\; \mathbf{D} \cdot (\mathbf{S}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j)\Big] \cdot \left[\frac{\mathbf{v}_1^\star - \mathbf{v}_1}{\mathbf{v}_2^\star - \mathbf{v}_2}\right] & \\ = \mathbf{D} \cdot \big(\mathsf{bin}(\mathbf{c}_M) - \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star)\big) \mod q. & \end{aligned} \tag{22}$$

The above implies that the vector

$$\begin{aligned} \mathbf{w} = \mathbf{S} \cdot (\mathbf{v}_1^\star - \mathbf{v}_1) + (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j) \cdot (\mathbf{v}_2^\star - \mathbf{v}_2) & \\ + \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star\big) - \mathsf{bin}(\mathbf{c}_M) & \end{aligned}$$

is a short integer vector of $\Lambda_q^{\perp}(\mathbf{D})$. Indeed, its norm can be bounded as $\|\mathbf{w}\| \leq \beta'' = \sqrt{2}(\ell + 2)\sigma^2 m^{3/2} + m^{1/2}$. We argue that it is non-zero with overwhelming probability. We already observed that $\mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star) - \mathsf{bin}(\mathbf{c}_M)$ is a non-zero vector of $\{-1,0,1\}^m$, which rules out the event that $(\mathbf{v}_1^\star,\mathbf{v}_2^\star)=(\mathbf{v}_1,\mathbf{v}_2)$. Hence, we can only have $\mathbf{w}=\mathbf{0}^m$ when the equality

$$\mathbf{S}\cdot(\mathbf{v}_1^\star-\mathbf{v}_1)+(\mathbf{S}_0+\sum_{j=1}^{\ell}\tau^{(i^\dagger)}[j]\cdot\mathbf{S}_j)\cdot(\mathbf{v}_2^\star-\mathbf{v}_2)$$
$$=\mathsf{bin}(\mathbf{c}_M)-\mathsf{bin}\big(\mathbf{D}_0\cdot\mathbf{s}^\star+\sum_{k=1}^{N}\mathbf{D}_k\cdot\mathfrak{m}_k^\star\big) \qquad (23)$$

holds over $\mathbb{Z}$. However, as long as either $\mathbf{v}_1^\star\neq\mathbf{v}_1$ or $\mathbf{v}_2^\star\neq\mathbf{v}_2$, the left-hand-side member of (23) is information theoretically unpredictable since the columns of matrices $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ are statistically hidden in the view of $\mathcal{A}$. Indeed, conditionally on the public key, each column of $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ has at least n bits of min-entropy, as shown by, e.g., [65, Lemma 2.7]. □

B.4 Proof of Theorem 2

Proof. The proof is very similar to the proof of Theorem 1 and we will only explain the changes.

Assuming that an adversary $\mathcal{A}$ can prove possession of a signature on a message $(\mathfrak{m}_1^\star,\ldots,\mathfrak{m}_N^\star)$ which has not been blindly signed by the issuer, we outline an algorithm $\mathcal{B}$ that solves a SIS instance $(\bar{\mathbf{A}},\beta)$, where $\bar{\mathbf{A}}=[\bar{\mathbf{A}}_1\mid\bar{\mathbf{A}}_2]\in U(\mathbb{Z}_q^{n\times 2m})$ with $\bar{\mathbf{A}}_1,\bar{\mathbf{A}}_2\in U(\mathbb{Z}_q^{n\times m})$.

At the outset of the game, $\mathcal{B}$ generates the common parameters par by choosing $\mathbf{B}\in_R\mathbb{Z}_q^{n\times m}$ and defining $\mathbf{G}_0=\mathbf{B}\cdot\mathbf{E}_0\in\mathbb{Z}_q^{n\times\ell}$, $\mathbf{G}_1=\mathbf{B}\cdot\mathbf{E}_1\in\mathbb{Z}_q^{n\times 2m}$. The short Gaussian matrices $\mathbf{E}_0\in\mathbb{Z}^{m\times\ell}$ and $\mathbf{E}_1\in\mathbb{Z}^{m\times 2m}$ are retained for later use. Also, $\mathcal{B}$ flips a coin $coin\in\{0,1,2\}$ as a guess for the kind of attack that $\mathcal{A}$ will mount. If $coin=0$, $\mathcal{B}$ expects a Type I forgery, where $\mathcal{A}$'s forgery involves a new $\tau^\star\in\{0,1\}^\ell$ that was never used by the signing oracle. If $coin=1$, $\mathcal{B}$ expects $\mathcal{A}$ to recycle a tag $\tau^\star$ involved in some signing query in its forgery. Namely, if $coin=1$, $\mathcal{B}$ expects an attack which is either a Type II forgery or a Type III forgery. If $coin=2$, $\mathcal{B}$ rather bets that $\mathcal{A}$ will break the soundness of the interactive argument systems used in the signature issuing protocol or the Prove protocol. Depending on the value of $coin\in\{0,1,2\}$, $\mathcal{B}$ generates the issuer's public key PK and simulates $\mathcal{A}$'s view in different ways.

- If $coin=0$, $\mathcal{B}$ undertakes to find a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}}_1)$, which in turn yields a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$. To this end, it defines $\mathbf{A}=\bar{\mathbf{A}}_1$ and generates PK by computing $\{\mathbf{A}_j\}_{j=0}^{\ell}$ as re-randomizations of $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$ as in the proof of Lemma 6. This implies that $\mathcal{B}$ can always answer signing queries using the trapdoor $\mathbf{T_C}\in\mathbb{Z}^{m\times m}$ of the matrix $\mathbf{C}$ without even knowing the messages hidden in the commitments $\mathbf{c}_\mathfrak{m}$ and $\{\mathbf{c}_k\}_{k=1}^{N}$, $\mathbf{c}_{s'}$. When the adversary generates a proof of possession of its own at the end of the game, $\mathcal{B}$ uses the matrices $\mathbf{E}_0\in\mathbb{Z}^{m\times\ell}$ and $\mathbf{E}_1\in\mathbb{Z}^{m\times 2m}$ as an extraction trapdoor to extract a plain message-signature pair $\big((\mathfrak{m}_1^\star,\ldots,\mathfrak{m}_N^\star),(\tau^\star,\mathbf{v}^\star,\mathbf{s}^\star)\big)$ from the ciphertexts

$\{\mathbf{c}_k^\star\}_{k=1}^N$ $(\mathbf{c}_{\mathbf{v}_1}^\star, \mathbf{c}_{\mathbf{v}_2}^\star)$, $\mathbf{c}_\tau^\star$, $\mathbf{c}_\mathbf{s}^\star$ produced by $\mathcal{A}$ as part of its forgery. If the extracted $\tau^\star$ is not a new tag, then $\mathcal{B}$ aborts. Otherwise, it can solve the given SIS instance exactly as in the proof of Lemma 6.

• If *coin* $= 1$, the proof proceeds as in the proof of Lemma 7 with one difference in Game 3. This difference is that Game 3 is no longer statistically indistinguishable from Game 2: instead, we rely on an argument based on the Rényi divergence. In Game 3, $\mathcal{B}$ generates PK exactly as in the proof of Lemma 7. This implies that $\mathcal{B}$ takes a guess $i^\dagger \leftarrow U(\{1,\ldots,Q\})$ with the hope that $\mathcal{A}$ will choose to recycle the tag $\tau^{(i^\dagger)}$ of the $i^\dagger$-th signing query (i.e., $\tau^\star = \tau^{(i^\dagger)}$). As in the proof of Lemma 7, $\mathcal{B}$ defines $\mathbf{D} = \bar{\mathbf{A}}_1 \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{A} = \bar{\mathbf{A}}_1 \cdot \mathbf{S}$ for a small-norm matrix $\mathbf{S} \in \mathbb{Z}^{m\times m}$ with Gaussian entries. It also "programs" the matrices $\{\mathbf{A}_j\}_{j=0}^\ell$ in such a way that the trapdoor precisely vanishes at the $i^\dagger$-th signing query: in other words, the sum

$$\mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\mathbf{A}_j = \bar{\mathbf{A}}_1 \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\cdot \mathbf{S}_j) + (h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\cdot h_j)\cdot \mathbf{C}$$

does not depend on the matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ (of which a trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m\times m}$ is known to $\mathcal{B}$) when $\tau^{(i)} = \tau^{(i^\dagger)}$, but it does for all other tags $\tau^{(i)} \neq \tau^{(i^\dagger)}$. In the setup phase, $\mathcal{B}$ also sets up a random matrix $\mathbf{D}_0 \in U(\mathbb{Z}_q^{2n\times 2m})$ which it obtains by choosing $\mathbf{A}' \hookleftarrow U(\mathbb{Z}_q^{n\times 2m})$ to define

$$\mathbf{D}_0 = \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{A}' \end{bmatrix} \in \mathbb{Z}_q^{2n\times 2m}. \qquad (24)$$

Then, it computes $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{s}_0 \in \mathbb{Z}_q^{2n}$ for a short Gaussian vector $\mathbf{s}_0 \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_0}$, which will be used in the $i^\dagger$-th query. Next, it samples short vectors $\mathbf{v}_1, \mathbf{v}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ to define

$$\mathbf{u} = \mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} - \mathbf{D}\cdot \mathsf{bin}(\mathbf{c}_M) \ \in \mathbb{Z}_q^n.$$

In addition, $\mathcal{B}$ picks extra small-norm matrices $\mathbf{R}_1,\ldots,\mathbf{R}_N \hookleftarrow \mathbb{Z}^{2m\times 2m}$ whose columns are sampled from $D_{\mathbb{Z}^m,\sigma}$, which are used to define randomizations of $\mathbf{D}_0$ by computing $\mathbf{D}_k = \mathbf{D}_0 \cdot \mathbf{R}_k$ for each $k \in \{1,\ldots,N\}$. The adversary is given public parameters $\mathsf{par} := \{\mathbf{B}, \mathbf{G}_0, \mathbf{G}_1, CK\}$, where $CK = \{\mathbf{D}_k\}_{k=0}^N$, and the public key $PK := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^\ell, \mathbf{D}, \mathbf{u}\big)$.

Using $\mathbf{T_C}$, $\mathcal{B}$ can perfectly emulate the signing oracle at all queries, except the $i^\dagger$-th query where the vector $\mathbf{s}''^{(i^\dagger)}$ chosen by $\mathcal{B}$ is sampled from a distribution that departs from $D_{\mathbb{Z}^{2m},\sigma_0}$. At the $i^\dagger$-th query, $\mathcal{B}$ uses the extraction trapdoor $\mathbf{E}_1 \in \mathbb{Z}^{m\times 2m}$ to obtain $\mathbf{s}'^{(i^\dagger)} \in \mathbb{Z}^{2m}$ and $\{\mathfrak{m}_k\}_{k=1}^N$ – which form a valid opening of $\mathbf{c}_\mathfrak{m}$ unless the soundness of the proof system is broken (note that the latter case is addressed by the situation *coin* $= 3$) – from the ciphertexts $\mathbf{c}_{s'}^{(i^\dagger)}$ and $\{\mathbf{c}_k\}_{k=1}^N$ sent by $\mathcal{A}$ at step 1 of the signing protocol. Then, $\mathcal{B}$ computes the vector $\mathbf{s}''^{(i^\dagger)}$ as

$$\mathbf{s}''^{(i^\dagger)} = \mathbf{s}_0 - \sum_{k=1}^{N} \mathbf{R}_k \cdot \mathfrak{m}_k^{(i^\dagger)} - \mathbf{s}'^{(i^\dagger)} \in \mathbb{Z}^{2m}, \qquad (25)$$

which satisfies $\mathbf{c}_M = \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} + \mathbf{D}_0 \cdot (\mathbf{s}'^{(i^\dagger)} + \mathbf{s}''^{(i^\dagger)})$ and allows returning $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}''^{(i^\dagger)})$ such that $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}'^{(i^\dagger)} + \mathbf{s}''^{(i^\dagger)})$ satisfies the verification equation of the signature scheme. Moreover, we argue that, with noticeable probability, the integer vector $\mathbf{s}^{(i^\dagger)} = \mathbf{s}'^{(i^\dagger)} + \mathbf{s}''^{(i^\dagger)}$ will be accepted by the verification algorithm since the Rényi divergence between the simulated distribution of $\mathbf{s}''^{(i^\dagger)}$ and its distribution in the real game will be sufficiently small. Indeed, its distribution is now that of a Gaussian vector $D_{\mathbb{Z}^{2m},\sigma_0,\mathbf{z}^\dagger}$ centered in

$$\mathbf{z}^\dagger = -\sum_{k=1}^{N} \mathbf{R}_k \cdot \mathfrak{m}_k^{(i^\dagger)} - \mathbf{s}'^{(i^\dagger)} \in \mathbb{Z}^{2m},$$

whose norm is at most $\|\mathbf{z}^\dagger\|_2 \leq N\sigma(2m)^{3/2} + \sigma(2m)^{1/2}$. By choosing the standard deviation σ_0 to be at least $\sigma_0 > N\sigma(2m)^{3/2} + \sigma(2m)^{1/2}$, the Rényi divergence between the simulated distribution of $\mathbf{s}''^{(i^\dagger)}$ (in Game 3) and its real distribution (which is the one of Game 2) can be kept constant: we have

$$R_2(\mathbf{s}''^{(i^\dagger),2} || \mathbf{s}''^{(i^\dagger),3}) \leq \exp\big(2\pi \cdot \frac{\|\mathbf{z}^\dagger\|_2^2}{\sigma_0^2}\big) \leq \exp(2\pi). \qquad (26)$$

This ensures that, with noticeable probability, $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}^{(i^\dagger)})$ will pass the verification test and lead $\mathcal{A}$ to eventually output a valid forgery. So, the success probability of $\mathcal{A}$ in Game 3 remains noticeable as (26) implies $\Pr[W_3] \geq \Pr[W_2]^2/\exp(2\pi)$.

When W_3 occurs in Game 3, $\mathcal{B}$ uses the matrices $(\mathbf{E}_0, \mathbf{E}_1)$ to extract a plain message-signature pair $\big((\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star), (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)\big)$ from the extractable commitments $\{\mathbf{c}_k^\star\}_{k=1}^{N}$ $(\mathbf{c}_{\mathbf{v}_1}^\star, \mathbf{c}_{\mathbf{v}_2}^\star)$, $\mathbf{c}_\tau^\star$, $\mathbf{c}_\mathbf{s}^\star$ generated by $\mathcal{A}$. At this point, two cases can be distinguished. First, if $\mathbf{c}_M \neq \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star + \mathbf{D}_0 \cdot \mathbf{s}^\star \bmod q$, then algorithm $\mathcal{B}$ can find a short vector of $\Lambda_q^\perp(\bar{\mathbf{A}}_1) = \Lambda_q^\perp(\mathbf{D})$ exactly as in the proof of Lemma 7. In the event that $\mathbf{c}_M = \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star + \mathbf{D}_0 \cdot \mathbf{s}^\star$, $\mathcal{B}$ can use the fact that the collision $\mathbf{c}_M = \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} + \mathbf{D}_0 \cdot \mathbf{s}^{(i^\dagger)}$ allows computing

$$\mathbf{w} = \mathbf{s}^\star - \mathbf{s}^{(i^\dagger)} + \sum_{k=1}^{N} \mathbf{R}_k \cdot \Big(\mathfrak{m}_k^\star - \mathfrak{m}_k^{(i^\dagger)}\Big) \ \in \mathbb{Z}^{2m},$$

which belongs to $\Lambda_q^\perp(\mathbf{D}_0)$ and has norm $\|\mathbf{w}\|_2 \leq N\sigma(2m)^{3/2} + 4\sigma_1 m^{3/2}$. Moreover, it is non-zero with overwhelming probability. Indeed, there exists at least one $k \in [1, N]$ such that $\mathfrak{m}_k^{(i^\dagger)} \neq \mathfrak{m}_k^\star$. Let us assume w.l.o.g. that they differ in their first two bits where $\mathfrak{m}_k^{(i^\dagger)}$ contains a 0 and $\mathfrak{m}_k^\star$ contains a 1 (recall that each bit b is encoded as $(\bar{b}, b)$ in both messages). This implies that $\mathbf{s}''^{(i^\dagger)}$ (as computed in (25)) does not depend on the first column of $\mathbf{R}_k$ but $\mathbf{w}$ does. Hence, given that the columns of $\mathbf{R}_k$ have at least n bits of min-entropy conditionally on $\mathbf{D}_k = \mathbf{D}_0 \cdot \mathbf{R}_k$, the vector $\mathbf{w} \in \mathbb{Z}^{2m}$ is unpredictable to the adversary.

Due to the definition of $\mathbf{D}_0 \in \mathbb{Z}_q^{2n \times 2m}$ in (24), we finally note that $\mathbf{w} \in \mathbb{Z}^{2m}$ is also a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$.

• If $coin = 2$, $\mathcal{B}$ faithfully generates par and PK, but it retains the extraction trapdoor $(\mathbf{E}_0, \mathbf{E}_1)$ associated with the dual Regev public keys $(\mathbf{G}_0, \mathbf{G}_1)$. Note that $\mathcal{A}$ can break the soundness of the proof system by either: (i) Generating ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$ and $\mathbf{c}_{s'}$ that do not encrypt an opening of $\mathbf{c}_{\mathfrak{m}}$ in the signature issuing protocol; (ii) Generating ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$, $\mathbf{c}_\tau$, $\mathbf{c}_{\mathbf{v}_1}$, $\mathbf{c}_{\mathbf{v}_2}$ and $\mathbf{c}_s$ that do not encrypt a valid signature in the Prove protocol. In either case, the reduction $\mathcal{B}$ is able to detect the event by decrypting dual Regev ciphertext using $(\mathbf{E}_0, \mathbf{E}_1)$ and create a breach in the soundness of the argument system.

It it easy to see that, since $coin \in \{0, 1, 2\}$ is chosen independently of $\mathcal{A}$'s view, it turns out to be correct with probability 1/3. As a consequence, if $\mathcal{A}$'s advantage is non-negligible, so is $\mathcal{B}$'s. □

B.5 Proof of Theorem 3

Proof. The proof is rather straightforward and consists of a sequence of three games.

Game 0: This is the real game. Namely, the adversary is given common public parameters par and comes up with a public key PK of its own. The adversary can run oblivious signing protocols with honest users. At each query, the adversary chooses a user index i and triggers an execution of the signing protocol with the challenger emulating the honest users. At some point, the adversary chooses some user index $i^\star$ for which the execution of the signing protocol ended successfully. At this point, the challenger $\mathcal{B}$ runs the real Prove protocol on behalf of user i. At the end of the game, the adversary outputs a bit $b' \in \{0, 1\}$. We define W_0 to be the event that $b' = 1$.

Game 1: This game is identical to Game 0 with the difference that, at each execution of the Prove protocol, the challenger runs the zero-knowledge simulator of the interactive proof system. The latter simulator uses either a trapdoor hidden in the common reference string (if Damgård's technique [37] is used) or proceeds by programming the random oracle which allows implementing the Fiat-Shamir heuristic. In either case, the statistical zero-knowledge property ensures that the adversary cannot distinguish Game 1 from Game 0 and $|\Pr[W_1] - \Pr[W_0]| \in \mathsf{negl}(\lambda)$.

Game 2: This game is like Game 1 except that, at each execution of the Prove protocol, the ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$, $\mathbf{c}_s$, $\mathbf{c}_\tau$, and $\mathbf{c}_{\mathbf{v}_1}$, $\mathbf{c}_{\mathbf{v}_2}$ encrypt random messages instead of the actual witnesses. The semantic security of the dual Regev cryptosystem ensures that, under the $\mathsf{LWE}_{n,q,chi}$ assumption, the adversary is unable to see the difference. Hence, we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda)$.

In Game 2, it is easy that the adversary is interacting with a simulator that emulates the user in the Prove protocol without using the any message-signature pair. We thus conclude that, under the $\mathsf{LWE}_{n,q,\chi}$ assumption, $\mathcal{A}$'s view cannot distinguish a real proof of signature possession from a simulated proof produced without any witness. □

C Security Proofs for the Dynamic Group Signature

C.1 Proof of Theorem 4

Proof. We prove that any adversary $\mathcal{A}$ with non-negligible success probability ϵ implies an algorithm $\mathcal{B}$ solving the SIS problem in the random oracle model.

Let $\mathcal{A}$ be such a PPT adversary. we build a PPT algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve $\mathsf{SIS}_{m,q,\beta}$: Specifically, $\mathcal{B}$ takes as input $\bar{\mathbf{A}} = \big[\bar{\mathbf{A}}_1 | \bar{\mathbf{A}}_2\big] \in \mathbb{Z}_q^{n \times 2m}$, where $\bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2 \in \mathbb{Z}_q^{n \times m}$, and finds $\mathbf{w} \in \varLambda_q^{\perp}(\bar{\mathbf{A}})$ with $0 < \|\mathbf{w}\| \leq \beta$.

Initialization. Algorithm $\mathcal{B}$ first chooses a random bit $coin \hookleftarrow U(\{0,1,2\})$ as a guess for the kind of misidentification attack that $\mathcal{A}$ will mount. Also, $\mathcal{B}$ chooses a random ℓ-bit string $\mathrm{id}^{\dagger} \hookleftarrow U(\{0,1\}^{\ell})$. In addition, $\mathcal{B}$ samples $i^{\star} \hookleftarrow U([1, Q_a])$.

Looking ahead, $coin = 0$ corresponds to the case where, after repeated executions of $\mathcal{A}$, the knowledge extractor of the proof system reveals witnesses containing a new identifier $\mathrm{id}^{\star} \in \{0,1\}^{\ell}$ that does not belong to any user in U^a. In this case, $\mathcal{B}$ will be able to exploit $\mathcal{A}$'s forgery when $\mathrm{id}^{\star} = \mathrm{id}^{\dagger}$. The case $coin = 1$ corresponds to $\mathcal{B}$'s expectation that the knowledge extractor will obtain the identifier $\mathrm{id}^{\star} = \mathrm{id}^{\dagger}$ of a group member in U^a (i.e., a group member that was legitimately introduced at the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query, for some $i^{\star} \in \{1, \ldots, Q_a\}$, where the identifier $\mathrm{id}^{\dagger}$ is used by $\mathcal{Q}_{\mathsf{a\text{-}join}}$), but $\mathsf{bin}(\mathbf{v}^{\star}) \in \{0,1\}^{2m}$ (which is encrypted in in $\mathbf{c}^{\star}_{\mathbf{v}_i}$ as part of the forgery $\varSigma^{\star}$) and the extracted $\mathbf{s}^{\star} \in \mathbb{Z}^{2m}$ are such that $\mathsf{bin}\big(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^{\star}) + \mathbf{D}_1 \cdot \mathbf{s}^{\star}\big) \in \{0,1\}^m$ does not match the string $\mathsf{bin}\big(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^{\star}}) + \mathbf{D}_1 \cdot \mathbf{s}_{i^{\star}}\big) \in \{0,1\}^{2m}$ for which user $i^{\star}$ obtained a membership certificate at the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query. When $coin = 1$, the choice of $i^{\star}$ corresponds to a guess that the knowledge extractor will reveal an ℓ-bit identifier that coincides with the identifier $\mathrm{id}^{\dagger}$ assigned to the user introduced at the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query. The last case $coin = 2$ corresponds to $\mathcal{B}$'s expectation that decrypting $\mathbf{c}^{\star}_{\mathbf{v}_i}$ (which is part of $\varSigma^{\star}$) and running the knowledge extractor on $\mathcal{A}$ will uncover vectors $\mathsf{bin}(\mathbf{v}^{\star}) \in \{0,1\}^{2m}$, $\mathbf{w}^{\star} \in \{0,1\}^m$ and $\mathbf{s}^{\star} \in \mathbb{Z}^{2m}$ such that $\mathbf{w}^{\star} = \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^{\star}) + \mathbf{D}_1 \cdot \mathbf{s}^{\star})$ and

$$\mathsf{bin}\big(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^{\star}) + \mathbf{D}_1 \cdot \mathbf{s}^{\star}\big) = \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^{\star}}) + \mathbf{D}_1 \cdot \mathbf{s}_{i^{\star}}\big) \tag{27}$$

but $(\mathsf{bin}(\mathbf{v}^{\star}), \mathbf{s}^{\star}) \neq (\mathsf{bin}(\mathbf{v}_{i^{\star}}), \mathbf{s}_{i^{\star}})$, where $\mathbf{v}_{i^{\star}} \in \mathbb{Z}_q^{4n}$ and $\mathbf{s}_{i^{\star}} \in \mathbb{Z}^{2m}$ are the vectors involved in the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query.

Depending on $coin \in \{0,1,2\}$, the group public key $\mathcal{Y}$ is generated using different methods.

• If $coin = 0$, algorithm $\mathcal{B}$ first randomly chooses $\mathrm{id}^{\dagger} \hookleftarrow U(\{0,1\}^{\ell})$ as a guess for the ℓ-bit string that will be revealed by the knowledge extractor of the proof system after repeated executions of the adversary $\mathcal{A}$. Then, it runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{T}_{\mathbf{C}}$ of $\varLambda_q^{\perp}(\mathbf{C})$ with $\|\widetilde{\mathbf{T}_{\mathbf{C}}}\| \leq \mathcal{O}(\sqrt{n \log q})$. Then, it chooses $\ell + 2$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_{\ell}, \mathbf{Q}_D \in \mathbb{Z}^{m \times m}$, each matrix having its columns sampled independently from $D_{\mathbb{Z}^m,\sigma}$. Then, $\mathcal{B}$ defines the matrices $\{\mathbf{A}_i\}_{i=0}^{\ell}$ as

$$\begin{cases} \mathbf{A}_0 = \bar{\mathbf{A}}_1 \cdot \mathbf{Q}_0 + (\sum_{i=1}^{\ell} \mathrm{id}^{\dagger}[i]) \cdot \mathbf{C} \\ \mathbf{A}_j = \bar{\mathbf{A}}_1 \cdot \mathbf{Q}_i + (-1)^{\mathrm{id}^{\dagger}[j]} \cdot \mathbf{C}, \qquad \text{for } j \in [1, \ell]. \\ \mathbf{D} = \bar{\mathbf{A}}_1 \cdot \mathbf{Q}_D \end{cases}$$

It also defines $\mathbf{A} = \bar{\mathbf{A}}_1$. Next, it samples a vector $\mathbf{e}_u \hookleftarrow D^m_{\mathbb{Z},\sigma}$ and computes a syndrome $\mathbf{u} = \bar{\mathbf{A}}_1 \cdot \mathbf{e}_u \in \mathbb{Z}_q^n$. It picks $\mathbf{D}_0, \mathbf{D}_1 \hookleftarrow U(\mathbb{Z}_q^{2n \times 2m})$ at random and also faithfully generates the GPV master key pair $(\mathbf{B}, \mathbf{T_B})$ as in Step 3 of the real setup algorithm. The group public key $\mathcal{Y} = \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{B}, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{F}, \mathbf{u}, \mathcal{OTS}, H, H_0\big)$ is finally given to $\mathcal{A}$.

Note that, for each $\mathrm{id} \neq \mathrm{id}^\dagger$, we have

$$\begin{aligned}\mathbf{A}_{\mathrm{id}} &= \Big[\bar{\mathbf{A}}_1 \;\Big|\; \mathbf{A}_0 + \textstyle\sum_{i=1}^{\ell} \mathrm{id}[i]\mathbf{A}_i\Big] \\ &= \Big[\bar{\mathbf{A}}_1 \;\Big|\; \bar{\mathbf{A}}_1 \cdot (\mathbf{Q}_0 + \textstyle\sum_{i=1}^{\ell} \mathrm{id}[i]\mathbf{Q}_i) + (\textstyle\sum_{i=1}^{\ell} \mathrm{id}^\dagger[i] + (-1)^{\mathrm{id}^\dagger[i]}\mathrm{id}[i]) \cdot \mathbf{C}\Big] \\ &= \big[\bar{\mathbf{A}}_1 \;\big|\; \bar{\mathbf{A}}_1 + h_{\mathrm{id}} \cdot \mathbf{C}\big] \qquad (28)\end{aligned}$$

where $h_{\mathrm{id}} \in [1, \ell]$ denotes the Hamming distance between the identifiers id and $\mathrm{id}^\dagger$. Since $q > \ell$, we have $h_{\mathrm{id}_j} \neq 0 \bmod q$ whenever $\mathrm{id}_j \neq \mathrm{id}^\dagger$, so that algorithm $\mathcal{B}$ is able to compute (see [3, Se. 4.2], using the basis $\mathbf{T_C}$ of $\Lambda_q^\perp(\mathbf{C})$ and the refined $\mathsf{GPVSample}$ of Lemma 2) a basis $\mathbf{T}_{\mathrm{id}}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}})$ with $\|\widetilde{\mathbf{T}_{\mathrm{id}}}\| \leq \Omega(\sqrt{n \log q \log n})$. In contrast, algorithm $\mathcal{B}$ lacks a trapdoor for $\mathbf{A}_{\mathrm{id}^\dagger}$ as the latter only depends on $\mathbf{A}$ and $\{\mathbf{Q}_k\}_{k=0}^{\ell}$. Observe that, since the columns of the matrices $\{\mathbf{Q}_k\}_{k=0}^{\ell}$ are sampled from $D_{\mathbb{Z}^m,\sigma}$, the matrices $\mathbf{A}_0, \ldots, \mathbf{A}_\ell$ are within statistical distance $2^{-\Omega(m)}$ of $U(\mathbb{Z}_q^{n \times m})$.

• If $coin = 1$, algorithm $\mathcal{B}$ sets up $\mathcal{Y}$ by defining $\mathbf{D} = \bar{\mathbf{A}}$. Initially, $\mathcal{B}$ chooses $Q_a - 1$ distinct strings $\mathrm{id}_1, \ldots, \mathrm{id}_{i^\star-1}, \mathrm{id}_{i^\star+1}, \ldots, \mathrm{id}_{Q_a} \in \{0,1\}^\ell$ such that, for each $i \in [1, Q_a] \backslash \{i^\star\}$, id_i will be embedded in the membership certificate returned in the i-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query. Let also $\mathrm{id}^\dagger = \mathrm{id}_{i^\star}$ be the ℓ-bit identifier that will be used in the $i^\star$-th query. The reduction $\mathcal{B}$ picks random $h_0, h_1, \ldots, h_\ell \in \mathbb{Z}_q$ under the constraints

$$\begin{aligned} h_{\mathrm{id}^\dagger} &= h_0 + \sum_{j=1}^{\ell} \mathrm{id}^\dagger[j] \cdot h_j = 0 \bmod q \\ h_{\mathrm{id}_i} &= h_0 + \sum_{j=1}^{\ell} \mathrm{id}_i[j] \cdot h_j \neq 0 \bmod q \qquad\qquad i \in \{1, \ldots, Q_a\} \setminus \{i^\dagger\}\end{aligned}$$

Next, $\mathcal{B}$ runs $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, $(\mathbf{D}_1, \mathbf{T}_{\mathbf{D}_1}) \leftarrow \mathsf{TrapGen}(1^{2n}, 1^{2m}, q)$ so as to obtain statistically random matrices $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{D}_1 \in \mathbb{Z}_q^{2n \times 2m}$ together with trapdoors $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$, $\mathbf{T}_{\mathbf{D}_1} \in \mathbb{Z}^{2m \times 2m}$ consisting of short bases of $\Lambda_q^\perp(\mathbf{C})$ and $\Lambda_q^\perp(\mathbf{D}_1)$, respectively. Then, $\mathcal{B}$ picks a random $\mathbf{D}_0 \hookleftarrow U(\mathbb{Z}_q^{2n \times 2m})$ and rerandomizes $\mathbf{D} = \bar{\mathbf{A}}_1 \in \mathbb{Z}_q^{n \times m}$ using Gaussian matrices $\mathbf{S}, \mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell \hookleftarrow \mathbb{Z}^{m \times m}$ whose columns are sampled from the distribution $D_{\mathbb{Z}^m,\sigma}$. Namely, from $\mathbf{D} = \bar{\mathbf{A}}_1$, $\mathcal{B}$ defines

$$\begin{aligned} \mathbf{A} &= \bar{\mathbf{A}}_1 \cdot \mathbf{S} \\ \mathbf{A}_0 &= \bar{\mathbf{A}}_1 \cdot \mathbf{S}_0 + h_0 \cdot \mathbf{C} \\ \mathbf{A}_j &= \bar{\mathbf{A}}_1 \cdot \mathbf{S}_j + h_j \cdot \mathbf{C} \qquad\qquad \forall j \in \{1, \ldots, \ell\}. \end{aligned} \qquad (29)$$

As part of the generation of $\mathcal{Y}$, the vector $\mathbf{u} \in \mathbb{Z}_q^n$ is obtained by picking short discrete Gaussian vectors $\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and computing

$$\mathbf{u} = [\mathbf{A} \ \mid \ \mathbf{A}_0 + \sum_{j=1}^{\ell} \mathrm{id}^\dagger[j]\mathbf{A}_j] \cdot \begin{bmatrix} \mathbf{d}_{i^\star,1} \\ \hline \mathbf{d}_{i^\star,2} \end{bmatrix} - \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M), \tag{30}$$

where $\mathbf{c}_M \hookleftarrow U(\mathbb{Z}_q^{2n})$ is a randomly chosen vector. Observe that, since $\mathbf{A}$ is statistically uniform over $\mathbb{Z}_q^{n \times m}$ and $\mathbf{d}_{i^\star,1} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, the distribution of $\mathbf{u}$ is statistically close to $U(\mathbb{Z}_q^n)$.

• If $coin = 2$, $\mathcal{B}$ picks $\bar{\mathbf{A}}' \hookleftarrow U(\mathbb{Z}_q^{n \times 2m})$ and a random matrix $\mathbf{Q} \hookleftarrow \mathbb{Z}^{2m \times 2m}$ whose columns are sampled from $D_{\mathbb{Z}^{2m},\sigma}$. These are used to define

$$\mathbf{D}_0 = \begin{bmatrix} \bar{\mathbf{A}} \\ \hline \bar{\mathbf{A}}' \end{bmatrix} \in \mathbb{Z}_q^{2n \times 2m},$$

and $\mathbf{D}_1 = \mathbf{D}_0 \cdot \mathbf{Q} \bmod q$, which is statistically close to $U(\mathbb{Z}_q^{2n \times 2m})$. All other components of $\mathcal{Y}$ are obtained by faithfully running the setup algorithm.

For each value of $coin \in \{0, 1, 2\}$, the group public key

$$\mathcal{Y} = \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{B}, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{F}, \mathbf{u}, \mathcal{OTS}, H, H_0\big)$$

has a distribution which is statistically close to that of the real scheme and $\mathcal{Y}$ is given to $\mathcal{A}$.

Queries. The reduction $\mathcal{B}$ starts interacting with the adversary $\mathcal{A}$ and the way it handles $\mathcal{A}$'s queries to the $\mathcal{Q}_{\mathsf{a\text{-}join}}$ oracle depends on the value of the bit $coin \in \{0, 1, 2\}$.

• If $coin = 0$, answers $\mathcal{Q}_{\mathsf{a\text{-}join}}$-queries as follows. When $\mathcal{A}$ triggers an execution of the joining protocol, it chooses a syndrome $\mathbf{v}_i \in \mathbb{Z}_q^n$. To answer the query, $\mathcal{B}$ chooses a fresh ℓ-bit identifier $\mathrm{id}_i \in \{0,1\}^\ell$ such that $\mathrm{id}_i \neq \mathrm{id}^\dagger$. If $\mathcal{A}$ also provides a correct signature sig_i such that $\mathrm{Verify}_{\mathsf{upk}[i]}(\mathbf{v}_i, sig_i) = 1$, $\mathcal{B}$ samples $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and uses the trapdoor $\mathbf{T_C}$ to compute a short vector $\mathbf{d}_i = [\mathbf{d}_{i,1}^T \mid \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}^{2m}$ such that

$$\mathbf{A}_{\mathrm{id}_i} \cdot \begin{bmatrix} \mathbf{d}_{i,1} \\ \hline \mathbf{d}_{i,2} \end{bmatrix} = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i\big), \tag{31}$$

where $\mathbf{A}_{\mathrm{id}_i} \in \mathbb{Z}_q^{n \times 2m}$ is the matrix in (28). Note that $\mathcal{B}$ is able to compute such a vector using the $\mathsf{SampleRight}$ algorithm of [3] (since the Hamming distance h_{id_i} between id_i and $\mathrm{id}^\star$ is non-zero). The membership certificate $\mathsf{cert}_i = (\mathrm{id}_i, \mathbf{d}_i, \mathbf{s}_i)$ is then returned to $\mathcal{A}$.

• If $coin = 1$, algorithm $\mathcal{B}$ responds each $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query depending on the index $i \in \{1, \ldots, Q_a\}$ of the query. Specifically, we distinguish two cases.

- If $i \neq i^\star$, $\mathcal{B}$ proceeds as in the previous case. Namely, it recalls the ℓ-bit identifier $\mathrm{id}_i \in \{0,1\}^\ell$ (for which $\mathrm{id}_i \neq \mathrm{id}^\dagger$) that was chosen in the setup phase and samples a short vector $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$. If $\mathcal{A}$ also provides a correct signature sig_i such that $\mathrm{Verify}_{\mathsf{upk}[i]}(\mathbf{v}_i, sig_i) = 1$, generates a membership certificate cert_i for $\mathcal{A}$ as in the case $coin = 0$. Note that

$$\begin{aligned} \mathbf{A}_{\mathrm{id}_i} &= \Big[\, \bar{\mathbf{A}} \cdot \mathbf{S} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{S}_0 + \textstyle\sum_{j=1}^{\ell} \mathrm{id}_i[j]\mathbf{S}_j) + h_{\mathrm{id}_i}\mathbf{C} \,\Big] \\ &= \big[\, \bar{\mathbf{A}} \cdot \mathbf{S} \;\big|\; \bar{\mathbf{A}} + h_{\mathrm{id}_i} \cdot \mathbf{C} \,\big] \end{aligned} \qquad (32)$$

  Since $h_{\mathrm{id}_i} \neq 0$, $\mathcal{B}$ can use the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ of $\Lambda_q^\perp(\mathbf{C})$ to compute a short vector $\mathbf{d}_i = [\mathbf{d}_{i,1}^T \mid \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}^{2m}$ such that

$$\mathbf{A}_{\mathrm{id}_i} \cdot \begin{bmatrix} \mathbf{d}_{i,1} \\ \mathbf{d}_{i,2} \end{bmatrix} = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}\big(\mathbf{D}_0 \cdot (\mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i\big),$$

  where $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$ is the syndrome chosen by $\mathcal{A}$ at step 1 of the joining protocol.
- If $i = i^\star$, $\mathcal{B}$ undertakes to generate a membership certificate $\mathsf{cert}_{i^\star}$ for the ℓ-bit identifier $\mathrm{id}^\dagger \in \{0,1\}^\ell$ that was chosen at the outset of the game. To this end, $\mathcal{B}$ has to compute $\mathsf{cert}_{i^\star}$ without using the trapdoor $\mathbf{T_C}$ since the matrix $\mathbf{A}_{\mathrm{id}^\dagger}$ does no longer depend on $\mathbf{C}$ in (32). This can be done by recalling the vector $\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2} \in \mathbb{Z}^m$ and $\mathbf{c}_M \in \mathbb{Z}_q^{2n}$ that were used to define $\mathbf{u} \in \mathbb{Z}_q^n$ in (30) and using $\mathbf{T}_{\mathbf{D}_1}$. If $\mathcal{A}$ provides a correct signature $sig_{i^\star}$ such that $\mathrm{Verify}_{\mathsf{upk}[i^\star]}(\mathbf{v}_{i^\star}, sig_{i^\star}) = 1$, $\mathcal{B}$ uses the trapdoor $\mathbf{T}_{\mathbf{D}_1}$ of $\Lambda_q^\perp(\mathbf{D}_1)$ to sample a short vector $\mathbf{s}_{i^\star} \in \mathbb{Z}^{2m}$ of $D_{\Lambda_q^{\mathbf{c}_{i^\star}}(\mathbf{D}_1),\sigma}$, where $\mathbf{c}_{i^\star} = \mathbf{c}_M - \mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^\star}) \bmod q$, satisfying

$$\mathbf{D}_1 \cdot \mathbf{s}_{i^\star} = \mathbf{c}_M - \mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^\star}) \mod q,$$

  before returning $\mathsf{cert}_{i^\star} = (\mathrm{id}^\dagger, \mathbf{d}_{i^\star} = [\mathbf{d}_{i^\star,1}^T \mid \mathbf{d}_{i^\star,2}^T]^T, \mathbf{s}_{i^\star})$ to $\mathcal{A}$. From the definition of $\mathbf{u} \in \mathbb{Z}_q^n$ (30), it is easy to see that $\mathsf{cert}_{i^\star} = (\mathrm{id}^\dagger, \mathbf{d}_{i^\star}, \mathbf{s}_{i^\star})$ forms a valid membership certificate for any membership secret $\mathbf{z}_{i^\star} \in \mathbb{Z}^{4m}$ corresponding to the syndrome $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}_{i^\star} \bmod q$.

Regardless of the value of $coin$, queries to the random oracle H are handled by returning a uniformly chosen value in $\{1,2,3\}^t$. For each $\kappa \leq Q_H$, we let r_κ denote the answer to the κ-th H-query. Of course, if the adversary makes a given query more than once, then $\mathcal{B}$ consistently returns the previously defined value. Queries to the random oracle H_0 are answered in the usual way, by returning a uniformly random value in the appropriate range.

Forgery. When $\mathcal{A}$ halts, it outputs a signature $\Sigma^\star = \big(\mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \pi_K^\star, sig^\star\big)$ on some message $M^\star$. At this point, $\mathcal{B}$ uses the trapdoor $\mathbf{T_B}$ to decrypt $\mathbf{c}_{\mathbf{v}_i}^\star$ and obtain an m-bit string $\mathsf{bin}(\mathbf{v}^\star) \in \{0,1\}^m$.

If we parse the proof of knowledge $\pi_K^\star$ as $(\{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t, \mathsf{Chall}_K^\star, \{\mathsf{Resp}_{K,j}^\star\}_{j=1}^t)$, with high probability, $\mathcal{A}$ must have invoked the random oracle H on the input $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$. Otherwise, the probability that $\mathsf{Chall}_K^\star = H(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ is negligible (at most 3^{-t}). It comes that, with probability at least $\varepsilon' := \epsilon - 3^{-t}$, the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}^\star_{\mathbf{v}_i}, \{\mathsf{Comm}^\star_{K,j}\}_{j=1}^t)$ coincides with the $\kappa^\star$-th random oracle query for some $\kappa^\star \leq Q_H$.

At this stage, the reduction $\mathcal{B}$ runs the adversary $\mathcal{A}$ up to $32 \cdot Q_H/(\epsilon - 3^{-t})$ times with the *same* random tape and input as in the initial run. All queries are answered as previously with one difference in the treatment of random oracle queries. Namely, the first $\kappa^\star - 1$ random oracle queries – which are identical to those of the first execution since $\mathcal{A}$ is run with the same random tape as before – receive the same answers $\mathsf{Chall}_1, \ldots, \mathsf{Chall}_{\kappa^\star-1}$ as in the first run. This implies that the $\kappa^\star$-th query will involve exactly the same tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}^\star_{\mathbf{v}_i}, \{\mathsf{Comm}^\star_{K,j}\}_{j=1}^t)$ as in the first run. However, from the $\kappa^\star$-th query onwards, $\mathcal{A}$ obtains fresh random oracle values $\mathsf{Chall}'_{\kappa^\star}, \ldots, \mathsf{Chall}'_{Q_H}$ at each new execution. The Improved Forking Lemma of Brickell *et al.* [24] guarantees that, with probability at least $1/2$, $\mathcal{B}$ can obtain a 3-fork involving the same tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}^\star_{\mathbf{v}_i}, \{\mathsf{Comm}^\star_{K,j}\}_{j=1}^t)$ with pairwise distinct answers $\mathsf{Chall}^{(1)}_{\kappa^\star}, \mathsf{Chall}^{(2)}_{\kappa^\star}, \mathsf{Chall}^{(3)}_{\kappa^\star} \in \{1,2,3\}^t$. With probability $1-(7/9)^t$ it can be shown that there exists an index $j \in \{1, \ldots, t\}$ for which the j-th bits of $\mathsf{Chall}^{(1)}_{\kappa^\star}, \mathsf{Chall}^{(2)}_{\kappa^\star}, \mathsf{Chall}^{(3)}_{\kappa^\star}$ are $(\mathsf{Chall}^{(1)}_{\kappa^\star,j}, \mathsf{Chall}^{(2)}_{\kappa^\star,j}, \mathsf{Chall}^{(3)}_{\kappa^\star,j}) = (1,2,3)$. From the corresponding responses $(\mathsf{Resp}^\star_{K,j}{}^{(1)}, \mathsf{Resp}^\star_{K,j}{}^{(2)}, \mathsf{Resp}^\star_{K,j}{}^{(3)})$, $\mathcal{B}$ is able to extract witnesses $(\mathbf{d}_1^\star, \mathbf{d}_2^\star) \in \mathbb{Z}^m \times \mathbb{Z}^m$, $\mathrm{id}^\star \in \{0,1\}^\ell$ and $\mathbf{w}^\star \in \{0,1\}^m$ from the proof of knowledge $\pi_K^\star$ such that

$$\mathbf{A}_{\mathrm{id}^\star} \cdot \left[\begin{array}{c}\mathbf{d}_1^\star \\ \hline \mathbf{d}_2^\star\end{array}\right] = \mathbf{u} + \mathbf{D} \cdot \mathbf{w}^\star$$

$$\mathbf{w}^\star = \mathsf{bin}\big(\mathbf{D}_0 \cdot (\mathsf{bin}(\mathbf{v}^\star) + \mathbf{D}_1 \cdot \mathbf{s}^\star\big),$$

At this point, $\mathcal{B}$ aborts and declares failure in the following situations:

- *coin* $= 0$ but $\mathrm{id}^\star \in \{0,1\}^\ell$ is recycled from some output of the $\mathcal{Q}_{\mathsf{a\text{-}join}}$ oracle.
- *coin* $= 0$ and $\mathrm{id}^\star \neq \mathrm{id}^\dagger$.
- *coin* $= 1$ but $\mathrm{id}^\star \in \{0,1\}^\ell$ never appeared in a membership certificate returned by the $\mathcal{Q}_{\mathsf{a\text{-}join}}$ oracle.
- *coin* $= 1$ and $\mathrm{id}^\star \in \{0,1\}^\ell$ belongs to some user in U^a, but this user is not the one introduced at the $i^\star$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query (i.e., $i^\star \neq i^\dagger$ and $\mathrm{id}^\star \neq \mathrm{id}^\dagger$).
- *coin* $= 1$ and the knowledge extractor revealed vectors $\mathsf{bin}(\mathbf{v}^\star) \in \{0,1\}^{2m}$ and $\mathbf{s}^\star \in \mathbb{Z}^{2m}$ satisfying the collision (27), where $\mathsf{bin}(\mathbf{v}_{i^\star})$ and $\mathbf{s}_{i^\star}$ are the vectors involved in the $i^\star$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$ query.
- *coin* $= 2$ and the knowledge extraction yields vectors $\mathsf{bin}(\mathbf{v}^\star) \in \{0,1\}^{2m}$ and $\mathbf{s}^\star \in \mathbb{Z}^{2m}$ such that the collision (27) does not occur.

We call fail the event that one of the above situations occurs. Given that the choices of *coin* $\hookleftarrow U(\{0,1,2\})$ and $i^\star \hookleftarrow U([1,Q_a])$ are completely independent of $\mathcal{A}$'s view, the choice of *coin* is correct with probability $1/3$. If *coin* $= 0$, $\mathcal{B}$'s choice of $\mathrm{id}^\dagger \hookleftarrow U(\{0,1\}^\ell)$ is correct with probability $1/(N_{\mathsf{gs}} - Q_a) \geq 1/N_{\mathsf{gs}}$ and, when *coin* $= 1$, $\mathcal{B}$'s correctly guesses $i^\star \in [1, Q_a]$ with probability $1/Q_a$. We find

$$\Pr[\neg\mathsf{fail}] \geq \frac{1}{3 \cdot \max(N_{\mathsf{gs}}, Q_a)} = \frac{1}{3 \cdot N_{\mathsf{gs}}}.$$

Assuming that fail does not occur, $\mathcal{B}$ can solve the problem instance as follows.

- If $coin = 0$, we have $\mathrm{id}^\star = \mathrm{id}^\dagger$ and $\mathcal{B}$ knows a short vector $\mathbf{e}_u \in \mathbb{Z}^m$ such that $\mathbf{u} = \bar{\mathbf{A}}_1 \cdot \mathbf{e}_u \bmod q$. Hence, it can obtain a short integer vector

$$\mathbf{h} = \mathbf{d}_1^\star + \big(\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathrm{id}^\dagger[i]\mathbf{Q}_i\big) \cdot \mathbf{d}_2^\star - \mathbf{Q}_D \cdot \mathsf{bin}(\mathbf{v}^\star) - \mathbf{e}_u \in \mathbb{Z}^m$$

such that $\bar{\mathbf{A}}_1 \cdot \mathbf{h} = \mathbf{0}^m \bmod q$. Moreover, we have $\mathbf{h} \neq \mathbf{0}^m$ w.h.p. since the syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ statistically hides $\mathbf{e}_u \in \mathbb{Z}^m$ in $\Lambda_q^{\mathbf{u}}(\bar{\mathbf{A}}_1)$. Finally, the norm of $\mathbf{h}$ is at most $\|\mathbf{h}\|_2 \leq (\ell+1)\sigma^2 m^{3/2} + \sigma m^{1/2}(m+2)$. This implies that $(\mathbf{h}^T \mid \mathbf{0}^m)^T$ is a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$ and solves the initial SIS instance.

- If $coin = 1$, the extracted witnesses $(\mathbf{d}_1^\star, \mathbf{d}_2^\star, \mathbf{s}^\star, \mathrm{id}^\star)$ and the decrypted $\mathsf{bin}(\mathbf{v}^\star)$ satisfy $\mathrm{id}^\star = \mathrm{id}^\dagger$,

$$\mathbf{w}^\star = \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^\star) + \mathbf{D}_1 \cdot \mathbf{s}^\star) \neq \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^\star}) + \mathbf{D}_1 \cdot \mathbf{s}_{i^\star}) = \mathbf{w}_{i^\star}$$

(since $\neg\mathsf{fail}$ implies that the collision (27) did not occur if $coin = 1$) and

$$\big[\mathbf{A} \mid \mathbf{A}_0 \mid \mathbf{A}_1 \mid \ldots \mid \mathbf{A}_\ell \mid -\mathbf{D}\big] \cdot \left[\begin{array}{c} \mathbf{d}_1^\star \\ \hline \mathbf{d}_2^\star \\ \hline \mathrm{id}^\dagger[1]\mathbf{d}_2^\star \\ \hline \vdots \\ \hline \mathrm{id}^\dagger[\ell]\mathbf{d}_2^\star \\ \hline \mathbf{w}^\star \end{array}\right] = \mathbf{u} \bmod q. \qquad (33)$$

Since $\mathcal{B}$ already knew short vectors $(\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2}, \mathbf{w}_{i^\star}) \in \mathbb{Z}^m \times \mathbb{Z}^m \times \mathbb{Z}^m$ such that

$$\big[\mathbf{A} \mid \mathbf{A}_0 \mid \mathbf{A}_1 \mid \ldots \mid \mathbf{A}_\ell \mid -\mathbf{D}\big] \cdot \left[\begin{array}{c} \mathbf{d}_{i^\star,1}^\star \\ \hline \mathbf{d}_{i^\star,2}^\star \\ \hline \mathrm{id}^\dagger[1]\mathbf{d}_{i^\star,2}^\star \\ \hline \vdots \\ \hline \mathrm{id}^\dagger[\ell]\mathbf{d}_{i^\star,2}^\star \\ \hline \mathbf{w}_{i^\star} \end{array}\right] = \mathbf{u} \bmod q, \qquad (34)$$

by subtracting (34) from (33), we find that

$$\mathbf{h} = \quad \mathbf{S} \cdot (\mathbf{d}_1^\star - \mathbf{d}_{i^\star,1}) + (\mathbf{S}_0 + \sum_{j=1}^{\ell} \mathrm{id}^\dagger[j]\mathbf{S}_j) \cdot (\mathbf{d}_2^\star - \mathbf{d}_{i^\star,2}) \; + (\mathbf{w}^\star - \mathbf{w}_{i^\star}) \qquad (35)$$

is a small-norm vector $\mathbf{h} \in \mathbb{Z}^m$ satisfying $\bar{\mathbf{A}}_1 \cdot \mathbf{h} = \mathbf{0} \bmod q$. We claim that $\mathbf{h} \neq \mathbf{0}$ with high probability. Indeed, we know that $\mathbf{w}^\star \neq \mathbf{w}_{i^\star}$ if $\neg\mathsf{fail}$ occurs. This implies that the last term of (35) is non-zero, which rules out that $(\mathbf{d}_1^\star, \mathbf{d}_2^\star) = (\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2})$. Since the columns of $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ have a lot of entropy conditionally on $\mathcal{Y}$, this implies that we can only have $\mathbf{h} = \mathbf{0}^m$ with negligible probability. Furthermore, the norm of $\mathbf{h}$ can be bounded by $\|\mathbf{h}\|_2 \leq 4\sigma^2 m^{3/2}(\ell + 2) + 2m^{1/2}$, so that $(\mathbf{h}^T \mid \mathbf{0}^m)^T$ solves the original SIS instance.

- If $coin = 2$, $\mathcal{B}$ is done as well since the collision (27) directly provides a vector

$$\mathbf{h} = \mathsf{bin}(\mathbf{v}^\star) - \mathsf{bin}(\mathbf{v}_i^\star) + \mathbf{Q} \cdot (\mathbf{s}^\star - \mathbf{s}_i^\star) \in \mathbb{Z}^{2m}$$

of $\Lambda_q^\perp(\mathbf{D}_0)$ (which is also in the lattice $\Lambda_q^\perp(\bar{\mathbf{A}})$ by construction) and has norm $\|\mathbf{h}\|_2 \leq 2(\sigma^2(2m)^{3/2} + (2m)^{1/2})$. Moreover, $\mathbf{h} \in \mathbb{Z}^{2m}$ is non-zero with overwhelming probability given that $\mathsf{bin}(\mathbf{v}^\star) \neq \mathsf{bin}(\mathbf{v}_i^\star)$ and the large amount of entropy retained by the columns $\mathbf{Q} \in \mathbb{Z}^{2m \times 2m}$ given $\mathbf{D}_1 = \mathbf{D}_0 \cdot \mathbf{Q}$. □

C.2 Proof of Theorem 5

Proof. Let us assume that a PPT adversary $\mathcal{A}$ can create a forgery $(M^\star, \Sigma^\star)$ that opens to some honest user $i \in U^b$ who did not sign $M^\star$. In the random oracle model, we give a reduction $\mathcal{B}$ that uses $\mathcal{A}$ to solve an instance of the $\mathsf{SIS}_{4n,4m,q,\beta}$ problem: $\mathcal{B}$ takes as input $\bar{\mathbf{A}} \in \mathbb{Z}_q^{4n \times 4m}$ and finds a non-zero short vector $\mathbf{w} \in \Lambda_q^\perp(\bar{\mathbf{A}})$.

Algorithm $\mathcal{B}$ generates the group public key $\mathcal{Y}$ by faithfully running the real setup algorithm with the sole difference that, at step 2 of Setup, $\mathcal{B}$ defines $\mathbf{F} = \bar{\mathbf{A}} \in \mathbb{Z}_q^{4n \times 4m}$. However, the distribution of $\mathcal{Y}$ is as in the real scheme. As a result of having generated $\mathcal{Y}$ itself, $\mathcal{B}$ knows $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ and $\mathcal{S}_{\mathsf{OA}} = \mathbf{T_B}$. The adversary $\mathcal{B}$ is run on input of the group public key

$$\mathcal{Y} := \Big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{B},\ \mathbf{D},\ \mathbf{D}_0,\ \mathbf{D}_1,\ \mathbf{F} = \bar{\mathbf{A}},\ \mathbf{u},\ \Pi^{\mathsf{OTS}},\ H,\ H_0)\Big).$$

If $\mathcal{A}$ chooses to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ is able to reveal $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ and $\mathcal{S}_{\mathsf{OA}} = \mathbf{T_B}$. Then, $\mathcal{B}$ starts interacting with $\mathcal{A}$ as follows.

- Q_{keyGM}-queries: If $\mathcal{A}$ decides to corrupt the group manager, $\mathcal{B}$ hands the secret key $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ to $\mathcal{A}$.
- $Q_{\mathsf{b\text{-}join}}$-queries: At any time $\mathcal{A}$ can act as a corrupted group manager and introduce a new honest user i in the group by invoking the $Q_{\mathsf{b\text{-}join}}$ oracle. At each $Q_{\mathsf{b\text{-}join}}$-query, $\mathcal{B}$ faithfully runs $\mathsf{J}_{\mathsf{user}}$ on behalf of the honest user in an execution of Join protocol.
- Q_{pub}-queries: These can be answered as in the real game, by having the simulator return $\mathcal{Y}$.
- Q_{sig}-queries: When the adversary $\mathcal{A}$ requests user $i \in U^b$ to sign a message M, $\mathcal{B}$ first generates a one-time key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(n)$ to compute $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n \times 2m}$. Next, $\mathcal{B}$ recalls the vector $\mathbf{z}_i \in \mathbb{Z}^{4m}$ that was chosen to define the syndrome $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i$ at step 1 of the Join protocol as well as the identifier $\mathrm{id}_i \in \{0,1\}^\ell$ and the short vectors $(\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \mathbf{s}_i)$ that were supplied by $\mathcal{A}$ in an earlier $Q_{\mathsf{b\text{-}join}}$-query. It faithfully computes a signature by IBE-encrypting $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{2m}$ and using $(\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \mathbf{s}_i, \mathbf{z}_i, \mathbf{s}_i, \mathrm{id}_i)$ to compute a

witness indistinguishable proof $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^t, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^t)$. Finally, $\mathcal{B}$ computes a one-time signature $sig = \mathcal{S}(\mathsf{SK}, (\mathbf{c}_{\mathbf{v}_i}, \pi_K))$ and returns the signature $\Sigma = \big(\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig\big)$ to $\mathcal{A}$.

When $\mathcal{A}$ halts, it outputs a signature $\Sigma^\star = \big(\mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \pi_K^\star, sig^\star\big)$ for some message $M^\star$, which opens to $i^\star \in U^b$ although user $i^\star$ did not sign the message $M^\star$ at any time. Since $(M^\star, \Sigma^\star)$ supposedly frames user $i^\star$, the opening of $\Sigma^\star$ must reveal the m-bit string $\mathsf{bin}(\mathbf{v}_{i^\star}) \in \{0,1\}^m$. We note that the reduction $\mathcal{B}$ has recollection of a short vector $\mathbf{z}_{i^\star} \in \mathbb{Z}^{4m}$ (of norm $\|\mathbf{z}_{i^\star}\| < 2\sigma\sqrt{m}$) such that $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}_{i^\star} \bmod q$ which it chose when running $\mathsf{J}_{\mathsf{user}}$ on behalf of user $i^\star$ when this user was introduced in the group. Hence, $\mathcal{B}$ would be able to solve its given SIS instance if it had another short vector $\mathbf{z}' \in \mathbb{Z}^{4m}$ satisfying $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}' \bmod q$. To compute such a vector, $\mathcal{B}$ proceeds by replaying the adversary $\mathcal{A}$ sufficiently many times and applying the Improved Forking Lemma of Brickell *et al.* [24].

If we parse $\pi_K^\star$ as $(\{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t, \mathsf{Chall}_K^\star, \{\mathsf{Resp}_{K,j}^\star\}_{j=1}^t)$, with high probability, $\mathcal{A}$ must have queried H on the input $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$. Otherwise, we would only have $\mathsf{Chall}_K^\star = H(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ with negligible probability 3^{-t}. It comes that, with probability at least $\varepsilon' := \varepsilon - 3^{-t}$, the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ was the input of the $\kappa^\star$-th random oracle query for some index $\kappa^\star \leq Q_H$.

At this point, the reduction $\mathcal{B}$ runs the adversary $\mathcal{A}$ up to $32 \cdot Q_H/(\varepsilon - 3^{-t})$ times with the *same* random tape and input as in the first run. All queries are answered as previously with one difference in the way to handle H-queries. Namely, the first $\kappa^\star - 1$ H-queries – which are the same as in the first execution since $\mathcal{A}$ is run with the same random tape – obtain the same answers $\mathsf{Chall}_1, \ldots, \mathsf{Chall}_{\kappa^\star - 1}$ as in the original run. This implies that the $\kappa^\star$-th query will also involve exactly the same tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ as in the original run. From the $\kappa^\star$-th query forward, however, the adversary $\mathcal{A}$ obtains fresh random oracle outputs $\mathsf{Chall}'_{\kappa^\star}, \ldots, \mathsf{Chall}'_{Q_H}$ at each new execution. The Improved Forking Lemma *et al.* [24] ensures that, with probability $> 1/2$, $\mathcal{B}$ obtains a 3-fork involving the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ of the initial run and with pairwise distinct answers $\mathsf{Chall}_{\kappa^\star}^{(1)}, \mathsf{Chall}_{\kappa^\star}^{(2)}, \mathsf{Chall}_{\kappa^\star}^{(3)} \in \{1,2,3\}^t$. Since the forgeries of the 3-fork all correspond to the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$, they open to the same m-bit string $\mathsf{bin}(\mathbf{v}_{i^\star}) \in \{0,1\}^m$ and which is uniquely determined by $\mathbf{c}_{\mathbf{v}}^\star$. In turn, this implies that the three forgeries all reveal the same $\mathsf{bin}(\mathbf{v}_{i^\star})$ at the second step of Open. With probability $1 - (7/9)^t$ it can be shown that there exists $j \in \{1, \ldots, t\}$ such that the j-th bits of $\mathsf{Chall}_{\kappa^\star}^{(1)}, \mathsf{Chall}_{\kappa^\star}^{(2)}, \mathsf{Chall}_{\kappa^\star}^{(3)}$ are $(\mathsf{Chall}_{\kappa^\star,j}^{(1)}, \mathsf{Chall}_{\kappa^\star,j}^{(2)}, \mathsf{Chall}_{\kappa^\star,j}^{(3)}) = (1,2,3)$. From the corresponding responses $({\mathsf{Resp}_{K,j}^\star}^{(1)}, {\mathsf{Resp}_{K,j}^\star}^{(2)}, {\mathsf{Resp}_{K,j}^\star}^{(3)})$, $\mathcal{B}$ is able to extract a short vector $\mathbf{z}' \in \mathbb{Z}^{4m}$ such that $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}' \bmod q$.

Due to the statistical witness indistinguishability of the Stern-like proof of knowledge which is used to generate signature, with overwhelming probability, we have $\mathbf{z}' \neq \mathbf{z}_{i^\star}$. Indeed, from the adversary's view, the distribution of $\mathbf{z}_{i^\star}$ is $D_{\Lambda_q^{\mathbf{v}_{i^\star}}(\mathbf{F}),\sigma}$, which means that it has at least n bits of min-entropy. Hence, the difference $\mathbf{h} = \mathbf{z}' - \mathbf{z}_{i^\star} \in \mathbb{Z}^{4m}$ is a suitably short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$. □

C.3 Proof of Theorem 6

Proof. We will proceed as in [62]: we will prove the anonymity through an hybrid argument using a sequence of indistinguishable games. The first game $\mathsf{Game}\ 0$ is the real experiment and the last one will be such that $\mathbf{Adv}_{\mathcal{A}}(\mathsf{Game}\ 6) = 0$

Game 0: This is the real anonymity game. The challenger runs algorithm $\mathsf{Setup}(1^\lambda, 1^{N_{\mathsf{gs}}})$ to obtain $(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, \mathcal{S}_{\mathsf{GM}})$ and then gives $\mathcal{Y}$ and $\mathcal{S}_{\mathsf{GM}}$ to the attacker $\mathcal{A}$. Using the secret key $\mathcal{S}_{\mathsf{OA}}$, the challenger is able to answer all the signatures opening queries. In the challenge phase, $\mathcal{A}$ sends a message $M^\star$ together with two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star), (\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, where $\mathsf{sec}_d^\star = \mathbf{z}_{i_d}^\star \in \mathbb{Z}^m$ and $\mathsf{cert}_d^\star = (\mathrm{id}_{i_d}^\star, \mathbf{d}_{i_d}^\star, \mathbf{s}_{i_d}^\star) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$ for each $d \in \{0,1\}$. The challenger sends back a challenge signature $\Sigma^\star = (\mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \pi_K^\star, sig^\star) \leftarrow \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_b^\star, \mathsf{sec}_b^\star, M^\star)$ for a random $b \hookleftarrow \{0,1\}$. The adversary returns a bit b' and the experiment return 1 if $b = b'$ or 0 otherwise.

Game 1: In this experiment, we slightly modify Game 0: at the outset of the game, the challenger generates the one-time signature key pair $(\mathsf{VK}^\star, \mathsf{SK}^\star)$. During the game, if $\mathcal{A}$ requests the opening of a valid signature $\Sigma = (\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig)$ where $\mathsf{VK} = \mathsf{VK}^\star$, the challenger returns a random bit and abort. However, this event F_1 would contradict the strong unforgeability of the one-time signature Π^{OTS}. Indeed, before the challenge phase $\mathsf{VK}^\star$ is independent of $\mathcal{A}$'s view and the probability that $\mathsf{VK}^\star$ shows up in $\mathcal{A}$'s queries is negligible. Moreover, after seeing the challenge signature $\Sigma^\star$, if $\mathcal{A}$ comes up with a valid signature $\Sigma = (\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig)$ such that $\mathsf{VK} = \mathsf{VK}^\star$, then sig is a forged one-time signature, which defeats the strong unforgeability of Π^{OTS}. Therefore the probability $\Pr[F_1]$ that the challenger aborts in this experiment is negligible. From here on, we thus assume that $\mathcal{A}$'s opening queries for valid signatures do not include $\mathsf{VK}^\star$.

Game 2: In this game, we program the random oracle H_0 in the following way: at the beginning of the game, we choose a uniformly random matrix $\mathbf{G}_0^\star \in \mathbb{Z}_q^{n \times 2m}$ and set $H_0(\mathsf{VK}^\star) = \mathbf{G}_0^\star$. From the attacker's point of view, the distribution of $\mathbf{G}_0^\star$ is statistically close to the one in the real attack game, as in [43]. As for other queries, for each fresh H_0-queries on VK, the challenger samples small-norm matrices $\mathbf{E}_{0,\mathsf{VK}} \hookleftarrow D_{\mathbb{Z}^m,\sigma}^{2m}$ and programs the oracle such that $H_0(\mathsf{VK}) = \mathbf{B} \cdot \mathbf{E}_{0,\mathsf{VK}} \bmod q$. The chosen matrices $\mathbf{E}_{0,\mathsf{VK}}$ are retained for later use. Note that the values of $H_0(\mathsf{VK})$ are statistically close to the uniform. For any query involving a previously queried VK, the challenger consistently returns the previously stored images. The view of the attacker remains the same as in Game 1, as in the security proof of the GPV IBE [43].

Game 3: Here, we will change the behaviour of the opening algorithm. Namely, at each fresh oracle query, we still store the matrices $\mathbf{E}_{0,\mathsf{VK}} \in \mathbb{Z}_q^{m \times 2m}$ and, at the beginning of the game, the challenger samples an uniformly random $\mathbf{B}^\star \in \mathbb{Z}_q^{n \times m}$. When the adversary queries the opening of a signature $\Sigma = (\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i},\ \pi_K, sig)$, $\mathcal{B}$ recalls the small-norm matrices $\mathbf{E}_{0,\mathsf{VK}}$ which were defined when $\mathcal{A}$ first queried $H_0(\mathsf{VK})$. These matrices are used as "decryption matrices" to open Σ for the corresponding $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n \times 2m}$. Similarly to the security proof [43], the distribution of $\mathbf{G}_0$ is statistically close to the uniform, implying that Game 2 and Game 3 are statistically indistinguishable.

Game 4: Instead of faithfully generating the NIZKPoK π_K, the challenger simulates the proof without using the witness (note that this is possible since the HVZK property of the underlying proof system is preserved under parallel repetitions). This is done by running the simulator for the underlying interactive protocol for each $j \in 1, \ldots, t$, and then programming the random oracle H accordingly. The challenge signature $\Sigma^\star = (\mathsf{VK}^\star, \mathbf{c}^\star_{\mathbf{v}_i}, \pi^\star_K, sig^\star)$ is statistically close to the challenge signature of the previous game, because the proof system is statistically zero-knowledge. Consequently, Game 3 and Game 4 are indistinguishable.

Game 5: In this game, we change the generation of the ciphertext $\mathbf{c}^\star_{\mathbf{v}_i}$ in the challenge phase. Instead of using the real encryption algorithm of the GPV IBE to compute $\mathbf{c}^\star_{\mathbf{v}_i}$, we return truly random ciphertexts. In other words, we let

$$\mathbf{c}^\star_{\mathbf{v}_i} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 + \mathsf{bin}(\mathbf{v}^\star_{i_b})\lfloor q/2 \rfloor \end{pmatrix},$$

where $\mathbf{v}^\star_{i_b} = \mathbf{F} \cdot \mathbf{z}^\star_{i_b}$, and where $\mathbf{z}_1 \hookleftarrow U(\mathbb{Z}_q^m)$, $\mathbf{z}_2 \hookleftarrow U(\mathbb{Z}_q^{2m})$ are uniformly random. The hardness of the decisional $\mathsf{LWE}_{q,\alpha}$ problem implies that $\mathbf{c}^\star_{\mathbf{v}_i}$ in Game 4 and Game 5 are computationally indistinguishable. If $\mathcal{A}$ can distinguish between these two games, it can distinguish

$$\begin{pmatrix} \mathbf{B}^T \\ \hline \mathbf{G}_0^{\star T} \end{pmatrix} \mathbf{e}_0 + \begin{pmatrix} \mathbf{x}_1 \\ \hline \mathbf{x}_2 \end{pmatrix} \text{ from } \begin{pmatrix} \mathbf{z}_1 \\ \hline \mathbf{z}_2 \end{pmatrix},$$

which would break the decisional $\mathsf{LWE}_{q,\alpha}$ assumption.
Therefore, Game 4 and Game 5 are computationally indistinguishable.

Game $G^{(6)}$: Finally we make a conceptual modification on the previous game. Namely we sample uniformly random $\mathbf{z}'_1 \hookleftarrow U(\mathbb{Z}_q^m)$, $\mathbf{z}'_2 \hookleftarrow U(\mathbb{Z}_q^{2m})$ and assign

$$\mathbf{c}^\star_{\mathbf{v}_i} = \begin{pmatrix} \mathbf{z}'_1 \\ \mathbf{z}'_2 \end{pmatrix}.$$

Clearly, the distribution of $\mathbf{c}^\star_{\mathbf{v}_i}$ has not changed since Game 5. Since Game 6 does no longer depend on the challenger's bit $b \in \{0, 1\}$, the result follows. □

D Proof of Lemma 8

We first restate Lemma 8.

Lemma 10. *The protocol in Figure 1 is a statistical* ZKAoK *for the relation* $\mathrm{R}_{\mathrm{abstract}}$ *with perfect completeness, soundness error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(L \log q)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{P}, \mathbf{v})$*, outputs an accepted transcript which is statistically close to that produced by the real prover.*

- *There exists an efficient knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* Ch*, outputs* $\mathbf{x}' \in \mathsf{VALID}$ *such that* $\mathbf{P} \cdot \mathbf{x}' = \mathbf{v} \bmod q$.

Proof. Note that, by construction, the protocol is perfectly complete: If an honest prover follows the protocol, then he always gets accepted by the verifier. It is also easy to see that the communication cost is bounded by $\widetilde{\mathcal{O}}(L \log q)$.

We now will prove that the protocol is a statistical zero-knowledge argument of knowledge for the relation $\mathrm{R}_{\mathrm{abstract}}$.

Zero-Knowledge Property. We construct a PPT simulator SIM interacting with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$, such that, given only the public input, SIM outputs with probability negligibly close to $2/3$ a simulated transcript that is statistically close to the one produced by the honest prover in the real interaction.

The simulator first chooses a random $\overline{Ch} \in \{1, 2, 3\}$. This is a prediction of the challenge value that $\widehat{\mathcal{V}}$ will *not* choose.

Case $\overline{Ch} = 1$: Using basic linear algebra over $\mathbb{Z}_q$, SIM computes a vector $\mathbf{x}' \in \mathbb{Z}_q^L$ such that $\mathbf{P} \cdot \mathbf{x}' = \mathbf{v} \bmod q$. Next, it samples $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$C_1' = \mathsf{COM}(\pi, \mathbf{P} \cdot \mathbf{r}; \rho_1), \;\; C_2' = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2), \;\; C_3' = \mathsf{COM}(T_\pi(\mathbf{x}' + \mathbf{r}); \rho_3).$$

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

- If $Ch = 1$: Output $\perp$ and abort.
- If $Ch = 2$: Send $\mathrm{RSP} = \big(\pi, \mathbf{x}' + \mathbf{r}, \rho_1, \rho_3\big)$.
- If $Ch = 3$: Send $\mathrm{RSP} = \big(\pi, \mathbf{r}, \rho_1, \rho_2\big)$.

Case $\overline{Ch} = 2$: SIM samples $\mathbf{x}' \hookleftarrow U(\mathsf{VALID})$, $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$C_1' = \mathsf{COM}(\pi, \mathbf{P} \cdot \mathbf{r}; \rho_1), \;\; C_2' = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2), \;\; C_3' = \mathsf{COM}(T_\pi(\mathbf{x}' + \mathbf{r}); \rho_3).$$

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

- If $Ch = 1$: Send $\mathrm{RSP} = \big(T_\pi(\mathbf{x}'), T_\pi(\mathbf{r}), \rho_2, \rho_3\big)$.
- If $Ch = 2$: Output $\perp$ and abort.
- If $Ch = 3$: Send $\mathrm{RSP} = \big(\pi, \mathbf{r}, \rho_1, \rho_2\big)$.

Case $\overline{Ch} = 3$: SIM samples $\mathbf{x}' \hookleftarrow U(\mathsf{VALID})$, $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where $C_2' = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2)$, $C_3' = \mathsf{COM}(T_\pi(\mathbf{x}' + \mathbf{r}); \rho_3)$ as in the previous two cases, while

$$C_1' = \mathsf{COM}(\pi, \mathbf{P} \cdot (\mathbf{x}' + \mathbf{r}) - \mathbf{v}; \rho_1),$$

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

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

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

Argument of Knowledge. Suppose that $\mathrm{RSP}_1 = (\mathbf{t}_x, \mathbf{t}_r, \rho_2, \rho_3)$, $\mathrm{RSP}_2 = (\pi_2, \mathbf{y}, \rho_1, \rho_3)$, $\mathrm{RSP}_3 = (\pi_3, \mathbf{r}_3, \rho_1, \rho_2)$ are 3 valid responses to the same commitment $\mathrm{CMT} = (C_1, C_2, C_3)$, with respect to all 3 possible values of the challenge. The validity of these responses implies that:

$$\begin{cases} \mathbf{t}_x \in \mathsf{VALID}; \\ C_1 = \mathsf{COM}(\pi_2, \mathbf{P} \cdot \mathbf{y} - \mathbf{v}; \rho_1) = \mathsf{COM}(\pi_3, \mathbf{M} \cdot \mathbf{r}_3; \rho_1); \\ C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2) = \mathsf{COM}(T_{\pi_3}(\mathbf{r}_3); \rho_2); \\ C_3 = \mathsf{COM}(\mathbf{t}_x + \mathbf{t}_r; \rho_3) = \mathsf{COM}(T_{\pi_2}(\mathbf{y}); \rho_3). \end{cases}$$

Since COM is computationally binding, we can deduce that:

$$\mathbf{t}_x \in \mathsf{VALID}; \pi_2 = \pi_3; \mathbf{t}_r = T_{\pi_3}(\mathbf{r}_3); \mathbf{t}_x + \mathbf{t}_r = T_{\pi_2}(\mathbf{y}); \mathbf{P} \cdot \mathbf{y} - \mathbf{v} = \mathbf{P} \cdot \mathbf{r}_3 \bmod q.$$

Let $\mathbf{x}' = \mathbf{y} - \mathbf{r}_3$, then we have $T_{\pi_2}(\mathbf{x}') = \mathbf{t}_x \in \mathsf{VALID}$ which implies that $\mathbf{x}' \in \mathsf{VALID}$. Furthermore, we have $\mathbf{P} \cdot \mathbf{x}' = \mathbf{P} \cdot (\mathbf{y} - \mathbf{r}_3) = \mathbf{v} \bmod q$.

This concludes the proof. □