Zero-Knowledge Arguments for Matrix-Vector Relations and Lattice-Based Group Encryption

Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, Huaxiong Wang

Zero-Knowledge Arguments for Matrix-Vector Relations and Lattice-Based Group Encryption

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

[1] École Normale Supérieure de Lyon, Laboratoire LIP (France)
[2] School of Physical and Mathematical Sciences, Nanyang Technological University (Singapore)

Abstract. Group encryption (GE) is the natural encryption analogue of group signatures in that it allows verifiably encrypting messages for some anonymous member of a group while providing evidence that the receiver is a properly certified group member. Should the need arise, an opening authority is capable of identifying the receiver of any ciphertext. As introduced by Kiayias, Tsiounis and Yung (Asiacrypt'07), GE is motivated by applications in the context of oblivious retriever storage systems, anonymous third parties and hierarchical group signatures. This paper provides the first realization of group encryption under lattice assumptions. Our construction is proved secure in the standard model (assuming interaction in the proving phase) under the Learning-With-Errors (LWE) and Short-Integer-Solution (SIS) assumptions. As a crucial component of our system, we describe a new zero-knowledge argument system allowing to demonstrate that a given ciphertext is a valid encryption under some hidden but certified public key, which incurs to prove quadratic statements about LWE relations. Specifically, our protocol allows arguing knowledge of witnesses consisting of $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ and a small-norm $\mathbf{e} \in \mathbb{Z}^m$ which underlie a public vector $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$ while simultaneously proving that the matrix $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$ has been correctly certified. We believe our proof system to be useful in other applications involving zero-knowledge proofs in the lattice setting.

Keywords. Lattices, zero-knowledge proofs, group encryption, anonymity.

1 Introduction

Since the pioneering work of Regev [50] and Gentry, Peikert and Vaikuntanathan (GPV) [24], lattice-based cryptography has been an extremely active research area. Not only do lattices enable powerful functionalities (e.g., [23,27]) that have no viable realizations under discrete-logarithm or factoring-related assumptions, they also offer a number of advantages over conventional number-theoretic techniques, like simpler arithmetic operations, their conjectured resistance to quantum attacks or a better asymptotic efficiency.

The design of numerous cryptographic protocols crucially relies on zero-knowledge proofs [26] to prove properties about encrypted or committed values

so as to enforce honest behavior on behalf of participants or protect the privacy of users. In the lattice settings, efficient zero-knowledge proofs are non-trivial to construct due to the limited amount of algebraic structure. While natural methods of proving knowledge of secret keys [45,43,32,41] are available, they are only known to work for specific languages. When it comes to proving circuit satisfiability, the best known methods are designed for the LPN setting [31] or take advantage of the extra structure available in the ring LWE setting [55,10]. Hence, these methods are not known to readily carry over to standard (i.e., non-ideal) lattices. In the standard model, the problem is even trickier as we do not have a lattice-based counterpart of Groth-Sahai proofs [29] and efficient non-interactive proof systems are only available for specific problems [49].

The difficulty of designing efficient zero-knowledge proofs for lattice-related languages makes it highly non-trivial to adapt privacy-preserving cryptographic primitives in the lattice setting. In spite of these technical hurdles, a recent body of work successfully designed anonymity-enabling mechanisms like ring signatures [32,2], blind signatures [51], group signatures [28,36,37,9,46,42,39] or, more recently, signature schemes with companion zero-knowledge protocols [38]. A common feature of all these works is that the zero-knowledge layer of the proposed protocols only deals with linear equations, where witnesses are only multiplied by public values.

In this paper, motivated by the design of advanced privacy-preserving protocols in the lattice setting, we construct zero-knowledge arguments for non-linear statements among witnesses consisting of vectors and matrices. For suitable parameters $q, n, m \in \mathbb{Z}$, we consider zero-knowledge argument systems whereby a prover can demonstrate knowledge of secret matrices $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$ and vectors $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathbb{Z}^m$ such that: (i) $\mathbf{e} \in \mathbb{Z}^m$ has small norm; (ii) A public vector $\mathbf{b} \in \mathbb{Z}_q^n$ equals $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$; (iii) The underlying pair $(\mathbf{X}, \mathbf{s})$ satisfies additional algebraic relations: for instance, it should be possible to prove possession of a signature on some representation of the matrix $\mathbf{X}$. In particular, our zero-knowledge argument makes it possible to prove that a given ciphertext is a well-formed LWE-based encryption with respect to some hidden, but certified public key. This protocol comes in handy in the design of *group encryption* schemes [34], where such languages naturally arise. In this paper, we thus construct the first construction of group encryption under lattice assumptions.

Group Encryption. As suggested by Kiayias, Tsiounis and Yung [34], group encryption (GE) is the encryption analogue of group signatures [19], which allow users to anonymously sign messages on behalf of an entire group they belong to. While group signatures aim at hiding the source of some message within a crowd administered by some group manager, group encryption rather seeks to hide its destination within a group of legitimate receivers. In both cases, a verifier should be convinced that the anonymous signer/receiver indeed belongs to a purported population. In order to keep users accountable for their actions, an opening authority (OA) is further empowered with some information allowing it to un-anonymize signatures/ciphertexts.

Kiayias, Tsiounis and Yung [34] formalized GE schemes as a primitive allowing the sender to generate publicly verifiable guarantees that: (1) The ciphertext is well-formed and intended for some registered group member who will be able to decrypt; (2) the opening authority will be able identify the receiver if necessary; (3) The plaintext satisfies certain properties such as being a witness for some public relation or the private key that underlies a given public key. In the model of Kiayias *et al.* [34], the message secrecy and anonymity properties are required to withstand active adversaries, which are granted access to decryption oracles in all security experiments.

As a natural application, group encryption allows a firewall to filter all incoming encrypted emails except those intended for some certified organization member and the content of which is additionally guaranteed to satisfy certain requirements, like the absence of malware.

GE schemes are also motivated by natural privacy applications such as anonymous trusted third parties, key recovery mechanisms or oblivious retriever storage systems. In optimistic protocols, GE allows verifiably encrypting messages to *anonymous* trusted third parties which mostly remain off-line and only come into play to sort out conflicts. In order to protect privacy-sensitive information such as users' citizenship, group encryption makes it possible to hide the identity of users' preferred trusted third parties within a set of properly certified trustees.

In cloud storage services, GE enables privacy-preserving asynchronous transfers of encrypted datasets. Namely, it allows users to archive encrypted datasets on remote servers while convincing those servers that the data is indeed intended for some anonymous certified client who paid a subscription to the storage provider. Moreover, a judge should be able to identify the archive's recipient in case a misbehaving server is found guilty of hosting suspicious transaction records or any other illegal content.

As pointed out by Kiayias *et al.* [34], group encryption also implies a form of hierarchical group signatures [54], where signatures can only be opened by a set of eligible trustees operating in a very specific manner determiner by the signer.

Related work. Kiayias, Tsiounis and Yung (KTY) [34] formalized the notion of group encryption and provided a modular design using zero-knowledge proofs, digital signatures, anonymous CCA-secure public-key encryption and commitment schemes. They also gave an efficient instantiation using Paillier's cryptosystem [47] and Camenisch-Lysyanskaya signatures [15].

Cathalo, Libert and Yung [18] designed a non-interactive system in the standard model under non-interactive pairing-related assumptions. El Aimani and Joye [3] suggested various efficiency improvements with both interactive and non-interactive proofs.

Libert *et al.* [40] empowered the GE primitive with a refined traceability mechanism akin to that of traceable signatures [33]. Namely, by releasing a user-specific trapdoor, the opening authority can allow anyone to publicly trace ciphertexts encrypted for this specific group member without affecting the privacy of other users. Back in 2010, Izabachène, Pointcheval and Vergnaud [30] considered the problem of eliminating subliminal channels in a different form of

traceable group encryption.

As a matter of fact, all existing realizations of group encryption or similar primitives rely on traditional number theoretic assumptions like the hardness of factoring or computing discrete logarithms. In particular, all of them are vulnerable to quantum attacks. For the sake of not putting all one's eggs in the same basket, it is highly desirable to have instantiations based on alternative, quantum-resistant foundations.

Our results and techniques. We put forth the first lattice-based realization of the group encryption primitive and prove its security under the Learning-With-Errors (LWE) [50] and Short-Integer-Solution (SIS) [4] assumptions. As in the original design of Kiayias, Tsiounis and Yung [34], the security analysis of our scheme stands in the standard model if we avail ourselves of interaction between the prover and the verifier. In the random oracle model [8], the Fiat-Shamir paradigm [22] readily provides a non-interactive solution based on the same hardness assumptions.

As a core ingredient of our GE scheme, we develop a new technique allowing to prove that a given ciphertext is a valid LWE-based encryption under some hidden but certified public key. Via a novel extension of Stern-like zero-knowledge arguments [53,32] in the lattice setting, we provide a method of proving quadratic relations between a secret certified matrix and a secret vector occurring in LWE-related languages. We believe our zero-knowledge arguments to be of independent interest as they find applications in other protocols involving zero-knowledge proofs in lattice-based cryptography.

It was shown by Kiayias *et al.* [34] that, in order to design a GE scheme, three ingredients are necessary: we need digital signatures, anonymous (i.e., key-private [7]) public-key encryption and zero-knowledge proofs. While the first two ingredients are available in lattice-based cryptography, suitable zero-knowledge proof systems are currently lacking. The underlying proof system should allow the sender to prove that the ciphertext is well-formed and is decryptable by some certified group member without betraying the latter's identity. Such statements typically involve equations of the form $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, for which given integers n, m, q and vector $\mathbf{b} \in \mathbb{Z}_q^m$, the prover has to demonstrate possession of a certified matrix $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, vector $\mathbf{s} \in \mathbb{Z}_q^n$ and small-norm error vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying the equation. Existing mechanisms of proving relations appearing in lattice-based cryptosystems belong to two main classes. The first one, which uses "rejection sampling" techniques for Schnorr-like protocols [52], was introduced by Lyubashevsky [43]. The second class, which was initiated by Ling *et al.* [41], appeals to "decomposition-extension-permutation" techniques in lattice-based extensions [32] of Stern's protocol [53]. These techniques mainly deal with *linear equations*, where each term is a product of a public matrix with a secret vector, which possibly satisfies some additional constraints (e.g., smallness) to be proven. Here, we are presented with *quadratic equations* where some terms $\mathbf{X} \cdot \mathbf{s}$ are products of two secret witnesses $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{s} \in \mathbb{Z}_q^n$ which are involved in other equations. Proving such quadratic equations thus requires new ideas.

To overcome the above hurdle, we employ a divide-and-conquer strategy.

First, we consider the binary representations of $\mathbf{X}$ and $\mathbf{s}$, and view the product $\mathbf{X} \cdot \mathbf{s}$ as a bunch of bit-wise products $\{x_i \cdot s_j\}_{i,j}$. Now, although these bit-wise products still admit a quadratic nature, but to prove that each of them is well-formed, it suffices to demonstrate in zero-knowledge that $x_i \cdot s_j$ belongs to the set $B = \{0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1\}$ of cardinality 4. This can be done with a Stern-like sub-protocol, using the following extending-then-permuting technique. We first extend $x_i \cdot s_j$ to vector $\mathsf{ext}(x_i, s_j) \stackrel{\mathsf{def}}{=} (\overline{x}_i \cdot \overline{s}_j, \overline{x}_i \cdot s_j, x_i \cdot \overline{s}_j, x_i \cdot s_j)^\top \in \{0,1\}^4$ whose entries are elements of B (here, $\bar{c}$ denotes the bit $1 - c$). We then employ a special permutation, determined by two random bits b_x and b_s, to the entries of $\mathsf{ext}(x_i, s_j)$, such that the permuted vector is exactly the correct extension $\mathsf{ext}(x_i \oplus b_x, s_j \oplus b_s)$, where $\oplus$ denotes the addition modulo 2. Seeing that a permutation of $\mathsf{ext}(x_i, s_j)$ has entries in the set B, the verifier should be convinced that $x_i \cdot s_j \in B$. Meanwhile, the bits b_x and b_s act as one-time pads that perfectly hide x_i and s_j. Furthermore, to prove that the same bits x_i and s_j are involved in other equations, we establish similar extending-then-permuting mechanisms for their other appearances, and use the same one-time pads b_x and b_s, respectively, as those places.

Having settled the problem of proving quadratic relations, we are able to realize the desired zero-knowledge layer by combining our proof system with the techniques of [42,38]. These help us demonstrate possession of a signature on the user's public key while proving that this key is encrypted under the OA's public key. Since users' public keys consist of a matrix $\mathbf{B}_\mathsf{U} \in \mathbb{Z}_q^{n \times m}$, we actually encrypt a hash value of this matrix under the OA's public key while the sender proves knowledge of a signature on the binary decomposition of $\mathbf{B}_\mathsf{U}$. By using a suitable lattice-based hash function [25], the Stern-like protocols of [42,38] make it possible to prove that the hashed matrix encrypted under the OA's public key coincides with the one for which the sender knows a certificate and which served as a public key to encrypt the actual plaintext.

The last issue to sort out is to determine the appropriate encryption schemes to work with in the two public-key encryption components. The CCA2-secure cryptosystem implied by the Agrawal-Boneh-Boyen (ABB) identity-based encryption (IBE) scheme [1] via the CHK transformation [16] is a natural choice as it is one of the most efficient LWE-based candidates in the standard model. For technical reasons, we chose to use a variant of the ABB cryptosystem based on the trapdoor mechanism of Micciancio and Peikert [44] because it allows dispensing with zero-knowledge proofs of public key validity. Indeed, the Kiayias-Tsiounis-Yung model [34] mandates that certified public keys be valid public keys (for which an underlying private key exists). This requirement is easier to handle using Micciancio-Peikert trapdoors [44] since, unlike GPV trapdoors [24], they are guaranteed to exist for any public matrix.

2 Background and Definitions

2.1 Lattices

In our notations, all vectors are denoted in bold lower-case letters while 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 orthogonalization.

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

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*

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

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 ([6, 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 in [24], Gaussian distributions with lattice support can be sampled from efficiently, given a sufficiently short basis of the lattice.

Lemma 2 ([14, Le. 2.3]). *There exists a* PPT *(probabilistic polynomial-time) algorithm* 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 ([5, Th. 3.2]). *There exists a* PPT *algorithm* 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 $\mathbf{T}_\mathbf{A}$ 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 $\|\widetilde{\mathbf{T}_\mathbf{A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.*

Lemma 3 is often combined with the sampler from Lemma 2. Micciancio and Peikert [44] 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 TrapGen.

We rely on a basis delegation algorithm [17] which extends a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ into a trapdoor of any $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ whose left $n \times m$ submatrix is $\mathbf{A}$.

Lemma 4 ([17, Le. 3.2]). *There exists a* PPT *algorithm* 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* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$.

Like [13,11], we use a technique due to Agrawal, Boneh and Boyen [1] that realizes a punctured trapdoor mechanism [12]. Analogously to [44], we will use such a mechanism in the real scheme and not only in the proof.

Lemma 5 ([1, Th. 19]). *There exists a* PPT *algorithm* SampleRight *that takes as inputs matrices* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{C} \in \mathbb{Z}_q^{n \times \bar{m}}$*, a low-norm matrix* $\mathbf{R} \in \mathbb{Z}^{m \times \bar{m}}$*, a short basis* $\mathbf{T_C} \in \mathbb{Z}^{\bar{m} \times \bar{m}}$ *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 a short vector* $\mathbf{b} \in \mathbb{Z}^{m+\bar{m}}$ *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* $\Lambda_q^{\mathbf{u}}\big([\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C}]\big)$.

2.2 Computational Problems

The security of our schemes provably relies 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., [24, 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., [50,48,14]).

2.3 Syntax and Definitions of Group Encryption

We use the syntax and the security model of Kiayias, Tsiounis and Yung [34]. The group encryption (GE) primitive involves a sender, a verifier, a group manager (GM) that manages the group of receivers and an opening authority (OA) which is capable of identitying ciphertexts' recipients. In the syntax of [34], a

GE scheme is specified by the description of a relation $\mathcal{R}$ as well as a tuple $\mathsf{GE} = \big(\mathsf{SETUP}, \mathsf{JOIN}, \langle \mathcal{G}_r, \mathcal{R}, \mathsf{sample}_{\mathcal{R}} \rangle, \mathsf{ENC}, \mathsf{DEC}, \mathsf{OPEN}, \langle \mathcal{P}, \mathcal{V} \rangle\big)$ of algorithms or protocols. In details, SETUP is a set of initialization procedures that all take (implicitly or explicitly) a security parameter 1^λ as input. We call them $\mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$, $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$ and $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$. The first one of these procedures generates a set of public parameters param (like the KTY construction [34], we rely on a common reference string even when using interaction between provers and verifiers). The latter two procedures are used to produce key pairs $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$, $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$ for the GM and the OA. In the following, param is incorporated in the inputs of all algorithms although we sometimes omit to explicitly write it.

$\mathsf{JOIN} = (\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}})$ is an interactive protocol between the GM and the prospective user. After the execution of JOIN, the GM stores the public key pk and its certificate $\mathsf{cert}_{\mathsf{pk}}$ in a public directory $\mathsf{database}$. As in [35], we will restrict this protocol to have minimal interaction and consist of only two messages: the first one is the user's public key pk sent by $\mathsf{J}_{\mathsf{user}}$ to J_{GM} and the latter's response is a certificate $\mathsf{cert}_{\mathsf{pk}}$ for pk that makes the user's group membership effective. We do not require the user to prove knowledge of his private key sk or anything else about it. In our construction, valid keys will be publicly recognizable and users will not have to prove their validity. By avoiding proofs of knowledge of private keys, the security proof never has to rewind the adversary to extract those private keys, which allows supporting concurrent joins as advocated by Kiayias and Yung [35]. If applications demand it, it is possible to add proofs of knowledge of private keys in a modular way but our security proofs do not require rewinding the adversary in executions of JOIN.

Algorithm $\mathsf{sample}_{\mathcal{R}}$ allows sampling pairs $(x, w) \in \mathcal{R}$ (made of a public value x and a witness w) using keys $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$ produced by $\mathcal{G}_r(1^\lambda)$ which samples public/secret parameters for the relation $\mathcal{R}$. Depending on the relation, $\mathsf{sk}_{\mathcal{R}}$ may be the empty string (as in the scheme [34] and ours which both involve publicly samplable relations). The testing procedure $\mathcal{R}(x, w)$ uses $\mathsf{pk}_{\mathcal{R}}$ to return 1 whenever $(x, w) \in \mathcal{R}$. To encrypt a witness w such that $(x, w) \in \mathcal{R}$ for some public x, the sender fetches the pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ from $\mathsf{database}$ and runs the randomized encryption algorithm. The latter takes as input w, a label L, the receiver's pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ as well as public keys $\mathsf{pk}_{\mathsf{GM}}$ and $\mathsf{pk}_{\mathsf{OA}}$. Its output is a ciphertext $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$. On input of the same elements, the certificate $\mathsf{cert}_{\mathsf{pk}}$, the ciphertext Ψ and the random coins $coins_\Psi$ that were used to produce Ψ, the non-interactive algorithm $\mathcal{P}$ generates a proof π_Ψ that there exists a certified receiver whose public key was registered in $\mathsf{database}$ and who is able to decrypt Ψ and obtain a witness w such that $(x, w) \in \mathcal{R}$. The verification algorithm $\mathcal{V}$ takes as input Ψ, $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, π_Ψ and the description of $\mathcal{R}$ and outputs 0 or 1. Given Ψ, L and the receiver's private key sk, the output of DEC is either a witness w such that $(x, w) \in \mathcal{R}$ or a rejection symbol $\perp$. Finally, OPEN takes as input a ciphertext/label pair (Ψ, L) and the OA's secret key $\mathsf{sk}_{\mathsf{OA}}$ and returns a receiver's public key pk.

The model of [34] considers four properties termed correctness, message se-

curity, anonymity and soundness. In the security definitions, stateful oracles capture the adversary's interaction with the system. In the soundness game, the KTY model requires that pk belongs to the language of valid public keys. Here, we are implicitly assuming that the space of valid public keys is dense (all matrices are valid keys, as is the case in our scheme).

In the upcoming definitions, we sometimes use the notation $\langle \mathsf{output}_A | \mathsf{output}_B \rangle \leftarrow \langle A(\mathsf{input}_A), B(\mathsf{input}_B) \rangle (\mathsf{common\text{-}input})$ to denote the execution of a protocol between A and B obtaining their own outputs from their respective inputs.

Correctness. The correctness property requires that the following experiment returns 1 with overwhelming probability.

Experiment $\mathbf{Expt}^{\mathrm{correctness}}(\lambda)$

$\mathsf{param} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}) \leftarrow \mathcal{G}_r(\lambda)$; $(x, w) \leftarrow \mathsf{sample}_{\mathcal{R}}(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$;
$(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \leftarrow \mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
$\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert}_{\mathsf{pk}} | \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}} \rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}(\mathsf{sk}_{\mathsf{GM}}) \rangle (\mathsf{pk}_{\mathsf{GM}})$;
$\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$;
$\pi_\Psi \leftarrow \mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w, L, \Psi, coins_\Psi)$;
If $\big((w \neq \mathsf{DEC}(\mathsf{sk}, \Psi, L)) \vee (\mathsf{pk} \neq \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \Psi, L))$
$\vee (\mathcal{V}(\Psi, L, \pi_\Psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) = 0)\big)$ then return 0 else return 1;

Message Secrecy. The message secrecy property is defined by an experiment where the adversary has access to oracles that may be stateful (and maintain a state across queries) or stateless:

- $\mathsf{DEC}(\mathsf{sk})$: is a stateless oracle for the user decryption function DEC. When this oracle is restricted not to decrypt a ciphertext-label pair (Ψ, L), we denote it by $\mathsf{DEC}^{\neg\langle\Psi, L\rangle}$.
- $\mathsf{CH}^b_{\mathsf{ror}}(\lambda, \mathsf{pk}, w, L)$: is a real-or-random challenge oracle which is only called once. It returns $(\Psi, coins_\Psi)$ such that $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$ if $b = 1$ whereas, if $b = 0$, $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w', L)$ encrypts a random plaintext of length $O(\lambda)$ uniformly sampled in the plaintext space. In both cases, $coins_\Psi$ denote the random coins used to generate Ψ.
- $\mathsf{PROVE}^b_{\mathcal{P}, \mathcal{P}'}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, \mathsf{pk}_{\mathcal{R}}, x, w, \Psi, L, coins_\Psi)$: is a stateful oracle that can be invoked a polynomial number times. If $b = 1$, it replies by running the real prover $\mathcal{P}$ on the inputs to create an actual proof π_Ψ. If $b = 0$, the oracle runs a simulator $\mathcal{P}'$ that uses the same inputs as $\mathcal{P}$ except witness $w, coins_\Psi$ and generates a simulated proof.

These oracles are used in an experiment where the adversary controls the GM, the OA and all members except the honest receiver. The adversary $\mathcal{A}$ embodies the dishonest GM that certifies the honest receiver in an execution of JOIN. It is granted access to an oracle DEC which decrypts on behalf of that receiver. In the challenge phase, it transmits a state information aux to itself and invokes the challenge oracle for a label and a pair $(x, w) \in \mathcal{R}$ of its choice. After the challenge phase, it can also query the PROVE oracle many times and finally attempts to guess the challenger's bit b.

As pointed out in [34,18], designing an efficient simulator $\mathcal{P}'$ (for executing $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(.)$ when $b = 0$) is part of the security proof.

Definition 4. *A* GE *scheme satisfies message security if, for any PPT adversary $\mathcal{A}$, the experiment below returns* 1 *with probability at most* $1/2 + \mathsf{negl}(\lambda)$.

Experiment $\mathbf{Expt}^{\mathrm{sec}}_{\mathcal{A}}(\lambda)$

$\mathsf{param} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{aux}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) \leftarrow \mathcal{A}(\mathsf{param})$;
$\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert}_{\mathsf{pk}} | \mathsf{aux} \rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathcal{A}(\mathsf{aux}) \rangle (\mathsf{pk}_{\mathsf{GM}})$;
$(\mathsf{aux}, x, w, L, \mathsf{pk}_{\mathcal{R}}) \leftarrow \mathcal{A}^{\mathsf{DEC}(\mathsf{sk},.)}(\mathsf{aux})$; *If* $(x, w) \notin \mathcal{R}$ *then return* 0;
$b \hookleftarrow \{0,1\}$; $(\Psi, coins_\Psi) \leftarrow \mathsf{CH}^b_{\mathsf{ror}}(\lambda, \mathsf{pk}, w, L)$;
$b' \leftarrow \mathcal{A}^{\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(\mathsf{pk}_{\mathsf{GM}},\mathsf{pk}_{\mathsf{OA}},\mathsf{pk},\mathsf{cert}_{\mathsf{pk}},\mathsf{pk}_{\mathcal{R}},x,w,\Psi,L,coins_\Psi),\mathsf{DEC}^{\neg\langle\Psi,L\rangle}(\mathsf{sk},.)}(\mathsf{aux}, \Psi)$;
If $b = b'$ *then return* 1 *else return* 0;

Anonymity. In the experiment modeling the anonymity property, the adversary controls the entire system except the opening authority and two well-behaved users. The challenger thus introduces two honest users' public keys $\mathsf{pk}_0, \mathsf{pk}_1$ in $\mathsf{database}$ and thus obtains certificate for both $\mathsf{pk}_0, \mathsf{pk}_1$ from the adversarially-controlled GM. For a pair $(x, w) \in \mathcal{R}$ of its choice, the adversary obtains an encryption of w under pk_b for some $b \in \{0,1\}$ chosen by the challenger. The adversary is provided with decryption oracles w.r.t. both keys $\mathsf{pk}_0, \mathsf{pk}_1$. In addition, it has the following oracles at disposal:

- $\mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$: is a challenge oracle that is only queried once by the adversary. It returns a pair $(\Psi, coins_\Psi)$ consisting of a ciphertext $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b}, w, L)$ and the coin tosses $coins_\Psi$ that were used to generate Ψ.
- $\mathsf{USER}(\mathsf{pk}_{\mathsf{GM}})$: is a stateful oracle that obtains certificates from the adversary by simulating two executions of $\mathsf{J}_{\mathsf{user}}$ to introduce two honest users in the group. It uses a string keys where the outputs $(\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_{\mathsf{pk}_0})$, $(\mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_{\mathsf{pk}_1})$ of honest users are written as long as the adversarially-supplied certificates $\{\mathsf{cert}_{\mathsf{pk}_d}\}^1_{d=0}$ are valid w.r.t. $\mathsf{pk}_{\mathsf{GM}}$ (i.e., invalid certificates are ignored by the oracle and no entry is introduced in keys for them).
- $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, .)$: is a stateless oracle that simulates the opening algorithm and, on input of a GE ciphertext, returns the receiver's public key.

The reason why the USER oracle is needed is that both honest users' public keys $\mathsf{pk}_0, \mathsf{pk}_1$ must have been properly certified by the adversarially-controlled GM before the challenge phase because the adversary subsequently obtains proofs generated using $(\mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b})$.

Definition 5. *A* GE *scheme satisfies anonymity if, for any PPT adversary $\mathcal{A}$, the experiment below returns* 1 *with a probability not exceeding* $1/2 + \mathsf{negl}(\lambda)$.

Experiment $\mathbf{Expt}^{\text{anon}}_{\mathcal{A}}(\lambda)$

$\mathsf{param} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
$(\mathsf{aux}, \mathsf{pk}_{\mathsf{GM}}) \leftarrow \mathcal{A}(\mathsf{param}, \mathsf{pk}_{\mathsf{OA}})$; $\mathsf{aux} \leftarrow \mathcal{A}^{\mathsf{USER}(\mathsf{pk}_{\mathsf{GM}}), \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, .)}(\mathsf{aux})$;
If $\mathsf{keys} \neq (\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_{\mathsf{pk}_0}, \mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_{\mathsf{pk}_1})(\mathsf{aux})$ *then return* 0;
$(\mathsf{aux}, x, w, L, \mathsf{pk}_{\mathcal{R}}) \leftarrow \mathcal{A}^{\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, .), \mathsf{DEC}(\mathsf{sk}_0, .), \mathsf{DEC}(\mathsf{sk}_1, .)}(\mathsf{aux})$;
If $(x, w) \notin \mathcal{R}$ *then return* 0;
$b \hookleftarrow \{0,1\}$; $(\Psi, coins_\Psi) \leftarrow \mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$;
$b' \leftarrow \mathcal{A}^{\mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b}, x, w, \Psi, L, coins_\Psi,$
$\mathsf{OPEN}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_{\mathsf{OA}}, .), \mathsf{DEC}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_0, .), \mathsf{DEC}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_1, .))}(\mathsf{aux}, \Psi)$;
If $b = b'$ *then return* 1 *else return* 0;

Soundness. Here, the adversary creates the group of receivers by interacting with the honest GM. Its goal is to produce a ciphertext Ψ and a convincing proof that Ψ is valid w.r.t. a relation $\mathcal{R}$ of its choice but either: (1) The opening of Ψ reveals a receiver's public key pk that does not belong to any group member; (2) The output pk of OPEN is not a valid public key (*i.e.*, $\mathsf{pk} \notin \mathcal{PK}$, where $\mathcal{PK}$ is the language of valid public keys); (3) The ciphertext C is not in the space $\mathcal{C}^{x, L, \mathsf{pk}_{\mathcal{R}}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}}$ of valid ciphertexts. This notion is formalized by a game where the adversary is given access to a user registration oracle $\mathsf{REG}(\mathsf{sk}_{\mathsf{GM}}, .)$ that simulates J_{GM}. This oracle maintains a list $\mathsf{database}$ where registered public keys and their certificates are stored.

Definition 6. *A* GE *scheme is sound if, for any PPT adversary* $\mathcal{A}$*, the experiment below returns* 1 *with negligible probability.*

Experiment $\mathbf{Expt}^{\text{soundness}}_{\mathcal{A}}(\lambda)$

$\mathsf{param} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
$(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \leftarrow \mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$;
$(\mathsf{pk}_{\mathcal{R}}, x, \Psi, \pi_\Psi, L, \mathsf{aux}) \leftarrow \mathcal{A}^{\mathsf{REG}(\mathsf{sk}_{\mathsf{GM}}, .)}(\mathsf{param}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$;
If $\mathcal{V}(\Psi, L, \pi_\Psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) = 0$ *return* 0;
$\mathsf{pk} \leftarrow \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \Psi, L)$;
If $\big((\mathsf{pk} \notin \mathsf{database}) \vee (\mathsf{pk} \notin \mathcal{PK}) \vee (\Psi \notin \mathcal{C}^{x, L, \mathsf{pk}_{\mathcal{R}}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}})\big)$
then return 1 *else return* 0;

The model of Kiayias *et al.* [34] requires that pk belongs to the language of valid public keys, so that the adversary is considered to defeat the soundness property when (Ψ, L) opens to a key outside the language (i.e., $\mathsf{pk} \notin \mathcal{PK}$). In our scheme, we will assume that the space of valid public keys is dense in that all matrices of a given dimension are valid public keys, which have an underlying private key. We nevertheless use the same definition as [34] in order to emphasize that we are not relaxing the model in any way.

3 Warm-up: Decompositions, Extensions, Permutations

This section introduces the notations and techniques that will be used throughout the paper. Part of the covered material appeared (in slightly different forms)

in recent works [41,42,21,39,38] on Stern-like protocols [53]. The techniques that will be employed for handling quadratic relations (double-bit extension $\mathsf{ext}(\cdot,\cdot)$, expansion $\mathsf{expand}^{\otimes}(\cdot,\cdot)$ of matrix-vector product and the associated permuting mechanisms) are novel contributions of this paper.

3.1 Decompositions

For any $B \in \mathbb{Z}_+$, define the number $\delta_B := \lfloor \log_2 B \rfloor + 1 = \lceil \log_2(B+1) \rceil$ and the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \lfloor \frac{B+2^{j-1}}{2^j} \rfloor$, $\forall j \in [1, \delta_B]$. As observed in [41], the sequence satisfies $\sum_{j=1}^{\delta_B} B_j = B$ and any integer $v \in [0, B]$ can be decomposed into a binary vector $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^\top \in \{0,1\}^{\delta_B}$ such that $\sum_{j=1}^{\delta_B} B_j \cdot v^{(j)} = v$. We describe this decomposition procedure in a deterministic manner:

1. $v' := v$
2. For $j = 1$ to δ_B do:
   (i) If $v' \geq B_j$ then $v^{(j)} := 1$, else $v^{(j)} := 0$;
   (ii) $v' := v' - B_j \cdot v^{(j)}$.
3. Output $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^\top$.

Next, for any positive integers $\mathfrak{m}, B$, we define the decomposition matrix:

$$\mathbf{H}_{\mathfrak{m},B} := \begin{bmatrix} B_1 \ldots B_{\delta_B} & & & \\ & B_1 \ldots B_{\delta_B} & & \\ & & \ddots & \\ & & & B_1 \ldots B_{\delta_B} \end{bmatrix} \in \mathbb{Z}^{\mathfrak{m} \times \mathfrak{m}\delta_B}, \tag{1}$$

and the following injective functions:

(i) $\mathsf{vdec}_{\mathfrak{m},B} : [0,B]^{\mathfrak{m}} \to \{0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{v} = (v_1, \ldots, v_{\mathfrak{m}})^\top$ to vector $\left(\mathsf{idec}_B(v_1)^\top \| \ldots \| \mathsf{idec}_B(v_{\mathfrak{m}})^\top\right)^\top$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}_{\mathfrak{m},B}(\mathbf{v}) = \mathbf{v}$.

(ii) $\mathsf{vdec}'_{\mathfrak{m},B} : [-B,B]^{\mathfrak{m}} \to \{-1,0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{w} = (w_1, \ldots, w_{\mathfrak{m}})^\top$ to vector $\left(\sigma(w_1) \cdot \mathsf{idec}_B(w_1)^\top \| \ldots \| \sigma(w_{\mathfrak{m}}) \cdot \mathsf{idec}_B(w_{\mathfrak{m}})^\top\right)^\top$, where for each $i = 1, \ldots, \mathfrak{m}$: $\sigma(w_i) = 0$ if $w_i = 0$; $\sigma(w_i) = -1$ if $w_i < 0$; $\sigma(w_i) = 1$ if $w_i > 0$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}'_{\mathfrak{m},B}(\mathbf{w}) = \mathbf{w}$.

We also define the following matrix decomposition procedure. For positive integers n, m, q, define the injective function $\mathsf{mdec}_{n,m,q} : \mathbb{Z}_q^{m \times n} \to \{0,1\}^{mn\delta_{q-1}}$ that maps matrix $\mathbf{X} = [\mathbf{x}_1 | \ldots | \mathbf{x}_n] \in \mathbb{Z}_q^{m \times n}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$, to vector

$$\begin{aligned} \mathsf{mdec}_{n,m,q}(\mathbf{X}) &= \left(\mathsf{vdec}_{m,q-1}(\mathbf{x}_1)^\top \| \ldots \mathsf{vdec}_{m,q-1}(\mathbf{x}_n)^\top\right)^\top \\ &= (x_{1,1}, \ldots, x_{1,mk}, x_{2,1}, \ldots, x_{2,mk}, \ldots, x_{n,1}, x_{n,mk})^\top \in \{0,1\}^{nm\delta_{q-1}}, \end{aligned}$$

where, for each $(i,j) \in [n] \times [m\delta_{q-1}]$, $x_{i,j} \in \{0,1\}$ denotes the j-th bit of the decomposition of the i-th column of $\mathbf{X}$.

Looking ahead, when proving knowledge of witnesses $(\mathbf{X}, \mathbf{s}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$ satisfying $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, we will have to consider terms of the form $x_{i,j} \cdot s_{i,t}$, where $\mathbf{s} = (s_1, \ldots, s_n)^\top \in \mathbb{Z}_q^n$ and $(s_{i,1}, \ldots, s_{i,\delta_{q-1}})^\top = \mathsf{idec}_{q-1}(s_i)$ for each $i \in [n]$.

3.2 Extensions and Permutations

We now introduce the extensions and permutations which will be essential for proving quadratic relations.

- For each $c \in \{0,1\}$, denote by $\overline{c}$ the bit $1-c \in \{0,1\}$.
- For $c_1, c_2 \in \{0,1\}$, define the vector

$$\mathsf{ext}(c_1, c_2) = (\overline{c}_1 \cdot \overline{c}_2, \overline{c}_1 \cdot c_2, c_1 \cdot \overline{c}_2, c_1 \cdot c_2)^\top \in \{0,1\}^4.$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2} that transforms vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^\top \in \mathbb{Z}_q^4$ to vector $(v_{b_1,b_2}, v_{b_1,\overline{b}_2}, v_{\overline{b}_1,b_2}, v_{\overline{b}_1,\overline{b}_2})^\top$.
  Note that, for all $c_1, c_2, b_1, b_2 \in \{0,1\}$, we have the following:

$$\mathbf{z} = \mathsf{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathbf{z}) = \mathsf{ext}(c_1 \oplus b_1, c_2 \oplus b_2), \tag{2}$$

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

Now, for positive integers n, m, k, and for vectors

$$\mathbf{x} = (x_{1,1}, \ldots, x_{1,mk}, x_{2,1}, \ldots, x_{2,mk}, \ldots, x_{n,1}, x_{n,mk})^\top \in \{0,1\}^{nmk}$$

and $\mathbf{s}_0 = (s_{1,1}, \ldots, s_{1,k}, s_{2,1}, \ldots, s_{2,k}, \ldots, s_{n,1}, \ldots, s_{n,k})^\top \in \{0,1\}^{nk}$, we define the vector $\mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0) \in \{0,1\}^{4nmk^2}$ as

$$\begin{aligned}
\mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0) = \big(& \mathsf{ext}^\top(x_{1,1}, s_{1,1}) \| \mathsf{ext}^\top(x_{1,1}, s_{1,2}) \| \ldots \| \mathsf{ext}^\top(x_{1,1}, s_{1,k}) \| \\
& \| \mathsf{ext}^\top(x_{1,2}, s_{1,1}) \| \mathsf{ext}^\top(x_{1,2}, s_{1,2}) \| \ldots \| \mathsf{ext}^\top(x_{1,2}, s_{1,k}) \| \ldots \\
& \| \mathsf{ext}^\top(x_{1,mk}, s_{1,1}) \| \mathsf{ext}^\top(x_{1,mk}, s_{1,2}) \| \ldots \| \mathsf{ext}^\top(x_{1,mk}, s_{1,k}) \| \\
& \| \mathsf{ext}^\top(x_{2,1}, s_{2,1}) \| \mathsf{ext}^\top(x_{2,1}, s_{2,2}) \| \ldots \| \mathsf{ext}^\top(x_{2,1}, s_{2,k}) \| \ldots \\
& \| \mathsf{ext}^\top(x_{2,mk}, s_{2,1}) \| \mathsf{ext}^\top(x_{2,mk}, s_{2,2}) \| \ldots \| \mathsf{ext}^\top(x_{2,mk}, s_{2,k}) \| \ldots \\
& \| \mathsf{ext}^\top(x_{n,1}, s_{n,1}) \| \mathsf{ext}^\top(x_{n,1}, s_{n,2}) \| \ldots \| \mathsf{ext}^\top(x_{n,1}, s_{n,k}) \| \ldots \\
& \| \mathsf{ext}^\top(x_{n,mk}, s_{n,1}) \| \mathsf{ext}^\top(x_{n,mk}, s_{n,2}) \| \ldots \| \mathsf{ext}^\top(x_{n,mk}, s_{n,k}) \big)^\top.
\end{aligned}$$

That is, $\mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0)$ is obtained by applying ext to all pairs of the form $(x_{i,j}, s_{i,t})$ for $(i,j,t) \in [n] \times [mk] \times [k]$.

Now, for $\mathbf{b} = (b_{1,1}, \ldots, b_{1,mk}, b_{2,1}, \ldots, b_{2,mk}, \ldots, b_{n,1}, b_{n,mk})^\top \in \{0,1\}^{nmk}$ and $\mathbf{d} = (d_{1,1}, \ldots, d_{1,k}, d_{2,1}, \ldots, d_{2,k}, \ldots, d_{n,1}, \ldots, d_{n,k})^\top \in \{0,1\}^{nk}$, we define the permutation $P_{\mathbf{b},\mathbf{d}}$ that transforms vector

$$\begin{aligned}
\mathbf{v} = \big(& (\mathbf{v}_{1,1,1}^\top \| \ldots \| \mathbf{v}_{1,1,k}^\top) \| (\mathbf{v}_{1,2,1}^\top \| \ldots \| \mathbf{v}_{1,2,k}^\top) \| \ldots \| (\mathbf{v}_{1,mk,1}^\top \| \ldots \| \mathbf{v}_{1,mk,k}^\top) \| \\
& (\mathbf{v}_{2,1,1}^\top \| \ldots \| \mathbf{v}_{2,1,k}^\top) \| (\mathbf{v}_{2,2,1}^\top \| \ldots \| \mathbf{v}_{2,2,k}^\top) \| \ldots \| (\mathbf{v}_{2,mk,1}^\top \| \ldots \| \mathbf{v}_{2,mk,k}^\top) \| \\
& (\mathbf{v}_{n,1,1}^\top \| \ldots \| \mathbf{v}_{n,1,k}^\top) \| (\mathbf{v}_{n,2,1}^\top \| \ldots \| \mathbf{v}_{n,2,k}^\top) \| \ldots \| (\mathbf{v}_{n,mk,1}^\top \| \ldots \| \mathbf{v}_{n,mk,k}^\top) \big)^\top \in \mathbb{Z}^{4nmk^2},
\end{aligned}$$

consisting of nmk^2 blocks of length 4, to the vector $P_{\mathbf{b},\mathbf{d}}(\mathbf{v})$ of the form

$$\begin{aligned}
\big(& (\mathbf{w}_{1,1,1}^\top \| \ldots \| \mathbf{w}_{1,1,k}^\top) \| (\mathbf{w}_{1,2,1}^\top \| \ldots \| \mathbf{w}_{1,2,k}^\top) \| \ldots \| (\mathbf{w}_{1,mk,1}^\top \| \ldots \| \mathbf{w}_{1,mk,k}^\top) \| \\
& (\mathbf{w}_{2,1,1}^\top \| \ldots \| \mathbf{w}_{2,1,k}^\top) \| (\mathbf{w}_{2,2,1}^\top \| \ldots \| \mathbf{w}_{2,2,k}^\top) \| \ldots \| (\mathbf{w}_{2,mk,1}^\top \| \ldots \| \mathbf{w}_{2,mk,k}^\top) \| \\
& (\mathbf{w}_{n,1,1}^\top \| \ldots \| \mathbf{w}_{n,1,k}^\top) \| (\mathbf{w}_{n,2,1}^\top \| \ldots \| \mathbf{w}_{n,2,k}^\top) \| \ldots \| (\mathbf{w}_{n,mk,1}^\top \| \ldots \| \mathbf{w}_{n,mk,k}^\top) \big)^\top,
\end{aligned}$$

where for each $(i,j,t) \in [n] \times [mk] \times [k]$: $\mathbf{w}_{i,j,t} = T_{b_{i,j},d_{i,t}}(\mathbf{v}_{i,j,t})$.

Observe that, for all $\mathbf{b} \in \{0,1\}^{nmk}, \mathbf{d} \in \{0,1\}^{nk}$, we have:

$$\mathbf{z} = \mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0) \iff P_{\mathbf{b},\mathbf{d}}(\mathbf{z}) = \mathsf{expand}^{\otimes}(\mathbf{x} \oplus \mathbf{b}, \mathbf{s}_0 \oplus \mathbf{d}). \tag{3}$$

Next, we recall the notations, extensions and permutations used in previous Stern-like protocols [41,42,21,38] for proving linear relations.

For any positive integer t, denote by $\mathcal{S}_t$ the symmetric group of all permutations of t elements, by B_{2t} the set of all vectors in $\{0,1\}^{2t}$ having Hamming weight t, and by B_{3t} the set of all vectors in $\{-1,0,1\}^{3t}$ having exactly t coordinates equal to j, for each $j \in \{-1,0,1\}$. Note that for any $\phi \in \mathcal{S}_{2t}$ and $\psi \in \mathcal{S}_{3t}$, we have the following equivalences:

$$\mathbf{x} \in \mathsf{B}_{2t} \iff \phi(\mathbf{x}) \in \mathsf{B}_{2t} \quad \text{and} \quad \mathbf{y} \in \mathsf{B}_{3t} \iff \psi(\mathbf{y}) \in \mathsf{B}_{3t}. \tag{4}$$

The following extending procedures are defined for any positive integers t.

- $\mathsf{ExtendTwo}_t : \{0,1\}^t \to \mathsf{B}_{2t}$. On input vector $\mathbf{x}$ with Hamming weight w, it outputs $\mathbf{x}' = (\mathbf{x}^\top \| \mathbf{1}^{t-w} \| \mathbf{0}^w)^\top$.
- $\mathsf{ExtendThree}_t : \{-1,0,1\}^t \to \mathsf{B}_{3t}$. On input vector $\mathbf{y}$ containing n_j coordinates equal to j for $j \in \{-1,0,1\}$, output $\mathbf{y}' = (\mathbf{y}^\top \| \mathbf{1}^{t-n_1} \| \mathbf{0}^{t-n_0} \| (-\mathbf{1})^{t-n_{-1}})$.

We also use the following encodings and permutations to achieve fine-grained control over coordinates of binary witness-vectors.

- For any positive integer t, define the function encode_t that encodes vector $\mathbf{x} = (x_1, \ldots, x_t)^\top \in \{0,1\}^t$ to vector $\mathsf{encode}_t(\mathbf{x}) = (\bar{x}_1, x_1, \ldots, \bar{x}_t, x_t)^\top \in \{0,1\}^{2t}$.
- For any positive integer t and any vector $\mathbf{c} = (c_1, \ldots, c_t)^\top \in \{0,1\}^t$, define the permutation $F_{\mathbf{c}}^{(t)}$ that transforms vector $\mathbf{v} = (v_1^{(0)}, v_1^{(1)}, \ldots, v_t^{(0)}, v_t^{(1)})^\top \in \mathbb{Z}^{2t}$ into vector $F_{\mathbf{c}}^{(t)}(\mathbf{v}) = (v_1^{(c_1)}, v_1^{(\bar{c}_1)}, \ldots, v_t^{(c_t)}, v_t^{(\bar{c}_t)})^\top$.

Note that the following equivalence holds for all $t, \mathbf{c}$:

$$\mathbf{y} = \mathsf{encode}_t(\mathbf{x}) \iff F_{\mathbf{c}}^{(t)}(\mathbf{y}) = \mathsf{encode}_t(\mathbf{x} \oplus \mathbf{c}). \tag{5}$$

To close this warm-up section, we remark that the equivalences observed in (3), (4) and (5) will play crucial roles in our zero-knowledge layer.

4 The Supporting Zero-Knowledge Layer

In this section, we first demonstrate how to prove in zero-knowledge that a given vector $\mathbf{b}$ is a correct LWE evaluation, i.e., $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where the hidden matrix $\mathbf{X}$ and vector $\mathbf{s}$ may satisfy additional conditions. This sub-protocol, which we believe will have other applications, is one of the major challenges in our road towards the design of lattice-based group encryption. We then plug this building block into the big picture, and construct the supporting zero-knowledge argument of knowledge (ZKAoK) for our group encryption scheme (Section 5).

4.1 Proving the LWE Relation With Hidden Matrices

Let n, m, q, β be positive integers where $\beta \ll q$, and let $k = \delta_{q-1} = \lceil \log_2 q \rceil$. We identify $\mathbb{Z}_q$ as the set $\{0, 1, \ldots, q-1\}$. We consider a zero-knowledge argument system allowing prover $\mathcal{P}$ to convince verifier $\mathcal{V}$ on input $\mathbf{b} \in \mathbb{Z}_q^m$ that $\mathcal{P}$ knows secret matrix $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, and vectors $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in [-\beta, \beta]^m$ such that:

$$\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q. \tag{6}$$

Moreover, the argument system should be readily extended to proving that $\mathbf{X}$ and $\mathbf{s}$ satisfy additional conditions, such as:

- The bits representing $\mathbf{X}$ are certified by an authority, and the prover also knows that secret signature-certificate.
- The (secret) hash of $\mathbf{X}$ is correctly encrypted to a given ciphertext.
- The LWE secret $\mathbf{s}$ is involved in other linear equations.

Let $q_1, \ldots, q_k \in \mathbb{Z}_q$ be the sequence of integers obtained by decomposing $q-1$ using the technique recalled in Section 3.1, and define the row vector $\mathbf{g} = (q_1, \ldots, q_k)$. Let $\mathbf{X} = [\mathbf{x}_1 | \ldots | \mathbf{x}_n] \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{s} = (s_1, \ldots, s_n)^\top$. For each index $i \in [n]$, let us consider $\mathsf{vdec}_{m,q-1}(\mathbf{x}_i) = (x_{i,1}, \ldots, x_{i,mk})^\top \in \{0,1\}^{mk}$. Let $\mathsf{vdec}_{n,q-1}(\mathbf{s}) = (s_{1,1}, \ldots, s_{1,k}, s_{2,1}, \ldots, s_{2,k}, \ldots, s_{n,1}, \ldots s_{n,k})^\top \in \{0,1\}^{nk}$ and observe that $s_i = \mathbf{g} \cdot \mathsf{idec}_{q-1}(s_i) = \mathbf{g} \cdot (s_{i,1}, \ldots, s_{i,k})^\top$ for each $i \in [n]$. We have:

$$\begin{aligned}
\mathbf{X} \cdot \mathbf{s} &= \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i = \sum_{i=1}^{n} \mathbf{H}_{m,q-1} \cdot \mathsf{vdec}_{m,q-1}(\mathbf{x}_i) \cdot s_i \\
&= \mathbf{H}_{m,q-1} \cdot \Big(\sum_{i=1}^{n} (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^\top \Big) \bmod q.
\end{aligned}$$

Observe that, for each $i \in [n]$ and each $j \in [mk]$, we have

$$x_{i,j} \cdot s_i = x_{i,j} \cdot \mathbf{g} \cdot (s_{i,1}, \ldots, s_{i,k})^\top = (q_1, \ldots, q_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^\top.$$

We now extend vector $(q_1, q_2, \ldots, q_k)$ to $\mathbf{g}' = (0,0,0,q_1,0,0,0,q_2,\ldots,0,0,0,q_k) \in \mathbb{Z}_q^{4k}$. For all $(i,j) \in [n] \times [mk]$, we have:

$$x_{i,j} \cdot s_i = \mathbf{g}' \cdot (\mathsf{ext}^\top(x_{i,j}, s_{i,1}) \| \ldots \| \mathsf{ext}^\top(x_{i,j}, s_{i,k}))^\top.$$

Let us define the matrices

$$\mathbf{Q}_0 := \mathbf{I}_{mk} \otimes \mathbf{g}' = \begin{bmatrix} \mathbf{g}' & & & \\ & \mathbf{g}' & & \\ & & \ddots & \\ & & & \mathbf{g}' \end{bmatrix} \in \mathbb{Z}_q^{mk \times 4mk^2}, \tag{7}$$

and $\widehat{\mathbf{Q}} = [\overbrace{\mathbf{Q}_0 | \ldots | \mathbf{Q}_0}^{n \text{ times}}] \in \mathbb{Z}_q^{mk \times 4nmk^2}$. For each $i \in [n]$, define

$$\begin{aligned}
\mathbf{y}_i = (\ &\mathsf{ext}^\top(x_{i,1}, s_{i,1}) \| \ldots \| \mathsf{ext}^\top(x_{i,1}, s_{i,k}))^\top \| \mathsf{ext}^\top(x_{i,2}, s_{i,1} \| \ldots \| \mathsf{ext}^\top(x_{i,2}, s_{i,k}) \\
&\| \ldots \| \mathsf{ext}^\top(x_{i,mk}, s_{i,1} \| \ldots \| \mathsf{ext}^\top(x_{i,mk}, s_{i,k}))^\top \in \{0,1\}^{4mk^2}.
\end{aligned}$$

Then, for all $i \in [n]$, we have: $(x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^\top = \mathbf{Q}_0 \cdot \mathbf{y}_i$. Now, we note that

$$(\mathbf{y}_1^\top \| \ldots \| \mathbf{y}_n^\top)^\top = \mathsf{expand}^\otimes(\mathsf{mdec}_{n,m,q}(\mathbf{X}), \mathsf{vdec}_{n,q-1}(\mathbf{s})),$$

and

$$\sum_{i=1}^{n} (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^\top$$
$$= \sum_{i=1}^{n} \mathbf{Q}_0 \cdot \mathbf{y}_i = \widehat{\mathbf{Q}} \cdot \mathsf{expand}^\otimes(\mathsf{mdec}_{n,m,q}(\mathbf{X}), \mathsf{vdec}_{n,q-1}(\mathbf{s})). \tag{8}$$

Letting $\mathbf{Q} = \mathbf{H}_{m,q-1} \cdot \widehat{\mathbf{Q}} \in \mathbb{Z}_q^{m \times 4nmk^2}$ and left-multiplying (8) by $\mathbf{H}_{m,q-1}$, we obtain the equation:

$$\mathbf{X} \cdot \mathbf{s} = \mathbf{Q} \cdot \mathsf{expand}^\otimes\big(\mathsf{mdec}_{n,m,q}(\mathbf{X}), \mathsf{vdec}_{n,q-1}(\mathbf{s})\big) \bmod q.$$

This means that the task of proving knowledge of $(\mathbf{X}, \mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n \times [-\beta, \beta]^m$ such that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$ boils down to proving knowledge of $\mathbf{z} \in \{0,1\}^{4nmk^2}$, $\mathbf{x} \in \{0,1\}^{nmk}$, $\mathbf{s}_0 \in \{0,1\}^{nk}$ and a short $\mathbf{e} \in \mathbb{Z}^m$ such that

$$\mathbf{b} = \mathbf{Q} \cdot \mathbf{z} + \mathbf{I}_m \cdot \mathbf{e} \bmod q \qquad \text{and} \qquad \mathbf{z} = \mathsf{expand}^\otimes(\mathbf{x}, \mathbf{s}_0).$$

As the knowledge of small-norm $\mathbf{e}$ can easily be proved with Stern-like protocol (e.g., [41]), the challenging part is to prove in ZK the constraint of $\mathbf{z} = \mathsf{expand}^\otimes(\mathbf{x}, \mathbf{s}_0)$. To this end, we will use the following permuting technique inspired by the equivalence of equation (3). We sample uniformly random $\mathbf{d}_x \in \{0,1\}^{nmk}$ and $\mathbf{d}_s \in \{0,1\}^{nk}$, send $\mathbf{x}' = \mathbf{x} \oplus \mathbf{d}_x$ and $\mathbf{s}' = \mathbf{s}_0 \oplus \mathbf{d}_s$ to the verifier, and let the latter check that $P_{\mathbf{d}_x, \mathbf{d}_s}(\mathbf{z}) = \mathsf{expand}^\otimes(\mathbf{x}', \mathbf{s}')$. This will be sufficient to convince the verifier that the original vector $\mathbf{z}$ satisfies the required constraint. The crucial point is that no additional information about $\mathbf{x}$ and $\mathbf{s}_0$ is leaked, since these binary vectors are perfectly hidden under the "one-time pad" $\mathbf{d}_x$ and $\mathbf{d}_s$, respectively.

In the framework of Stern's protocol, the idea of using "one-time-pad" permutations further allows us to prove that $\mathbf{x}$ and $\mathbf{s}_0$ satisfy additional conditions, i.e., they appear in other equations. This is done by first setting up an equivalence similar to (3) in the places where these objects appear, and then, using the same "one-time pad" for each of them in all appearances. We will explain in detail how this technique can be realized in the next subsection.

4.2 The Main Zero-Knowledge Argument System

The zero-knowledge argument of knowledge used in our group encryption scheme (Section 5) will involve a system of 10 modular equations:

$$\begin{cases} \mathbf{v}_1 = \mathbf{M}_{1,1} \cdot \mathbf{w}_1 + \mathbf{M}_{1,2} \cdot \mathbf{w}_2 + \ldots + \mathbf{M}_{1,15} \cdot \mathbf{w}_{15} \bmod q, \\ \mathbf{v}_2 = \mathbf{M}_{2,1} \cdot \mathbf{w}_1 + \mathbf{M}_{2,2} \cdot \mathbf{w}_2 + \ldots + \mathbf{M}_{2,15} \cdot \mathbf{w}_{15} \bmod q, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mathbf{v}_{10} = \mathbf{M}_{10,1} \cdot \mathbf{w}_1 + \mathbf{M}_{10,2} \cdot \mathbf{w}_2 + \ldots + \mathbf{M}_{10,15} \cdot \mathbf{w}_{15} \bmod q, \end{cases} \tag{9}$$

where $\{\mathbf{M}_{i,j}\}_{(i,j)\in[10]\times[15]}$, $\{\mathbf{v}_i\}_{i\in[10]}$ are public matrices and vectors (which are possibly zero). Our goal is to prove knowledge of vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{15}$, such that (9) holds, and that these vectors have the following constraints.

1. $\mathbf{w}_1 \in \{0,1\}^{n\bar{m}k}$, $\mathbf{w}_2 \in \{0,1\}^{nk}$ and $\mathbf{w}_3 = \mathsf{expand}^{\otimes}(\mathbf{w}_1, \mathbf{w}_2) \in \{0,1\}^{4n\bar{m}k^2}$. (Note that these vectors are obtained via the techniques of Section 4.1.)
2. $\mathbf{w}_4, \mathbf{w}_5, \mathbf{w}_6, \mathbf{w}_7$ are $\{0,1\}$ vectors.
3. Vectors $\mathbf{w}_8, \ldots, \mathbf{w}_{14}$ have bounded infinity norms.
4. Vector $\mathbf{w}_{15}$ has the form $\left(\mathbf{d}_1^\top \| \mathbf{d}_2^\top \| \tau[1]\cdot\ \mathbf{d}_2^\top \| \ldots \| \tau[\ell]\cdot\ \mathbf{d}_2^\top\right)^\top$, for some vectors $\mathbf{d}_1, \mathbf{d}_2 \in [-\beta, \beta]^m$ and $\tau = (\tau[1], \ldots, \tau[\ell])^\top \in \{0,1\}^\ell$.

Towards achieving the goal, we employ a 4-step strategy.

1. The first step transforms all the secret vectors with infinity norm larger than 1 into vectors with infinity norm 1. This is done with the decomposition technique of Section 3.1.
2. The norm-1 vectors is then encoded or extended into vectors whose constraints are invariant under random permutations. This is done with the techniques described at the end of Section 3.2. The public matrices $\{\mathbf{M}_{i,j}\}_{i,j}$ are transformed accordingly to preserve the equations.
3. The third step unifies all the equations into one of the form $\mathbf{M}\cdot\mathbf{x} = \mathbf{v} \bmod q$, where $\mathbf{x}$ is a concatenation of the newly obtained witness-vectors.
4. In the final step, we run a Stern-like protocol to prove the unified equation $\mathbf{M}\cdot\mathbf{x} = \mathbf{v} \bmod q$, where a composed permutation is employed to prove the constraints of vector $\mathbf{x}$.

Our strategy subsumes the central ideas underlying recent works on Stern-like protocols [41,42,38] for lattice-based relations: preprocessing secret witness-vectors to make them provable-in-zero-knowledge with random permutations, unifying them into just one vector for the sake of convenience, and then running Stern's protocol in a classical manner.

The first step is applicable to vectors $\mathbf{w}_8, \ldots, \mathbf{w}_{14}$ and $\mathbf{w}_{15}$. Suppose that $\mathbf{w}_i$ has dimension m_i and infinity norm bound β_i, for $i \in [8, 14]$. Then we compute vector $\mathbf{w}'_i = \mathsf{vdec}_{m_i,\beta_i}(\mathbf{w}_i) \in \{-1,0,1\}^{m_i\delta_{\beta_i}}$. Note that $\mathbf{H}_{m_i,\beta_i}\cdot\mathbf{w}'_i = \mathbf{w}_i$. To decompose $\mathbf{w}_{15}$, we compute $\mathbf{d}'_j = \mathsf{vdec}_{m,\beta}(\mathbf{d}_j) \in \{-1,0,1\}^{m\delta_\beta}$, for $j = 1, 2$.

The second step performs the following encodings and extensions.

- Encode $\mathbf{w}_1$ and $\mathbf{w}_2$: Let $\mathbf{w}''_1 = \mathsf{encode}_{n\bar{m}k}(\mathbf{w}_1)$ and $\mathbf{w}''_2 = \mathsf{encode}_{nk}(\mathbf{w}_2)$. Note that to prove knowledge of $\mathbf{w}''_1$ and $\mathbf{w}''_2$, we will employ the "one-time pad" permuting technique implied by (5). The same one-time pads are used to prove that $\mathbf{w}_3 = \mathsf{expand}^{\otimes}(\mathbf{w}_1, \mathbf{w}_2)$, as discussed in Section 4.1.
- Extend vectors $\mathbf{w}_4, \ldots, \mathbf{w}_7, \mathbf{w}'_8, \ldots, \mathbf{w}'_{14}$ and $\mathbf{d}'_1, \mathbf{d}'_2, \tau$.
  For $i \in [4, 7]$, suppose that the binary vector $\mathbf{w}_i$ has dimension m_i. Then we extend it to $\mathbf{w}''_i = \mathsf{ExtendTwo}_{m_i}(\mathbf{w}_i) \in \mathsf{B}_{2m_i}$. For $i \in [8, 14]$, we extend $\mathbf{w}'_i$ to $\mathbf{w}''_i = \mathsf{ExtendThree}_{m_i\delta_{\beta_i}}(\mathbf{w}'_i) \in \mathsf{B}_{3m_i\delta_{\beta_i}}$. It follows from (4) that, the knowledge of vectors $\{\mathbf{w}''_i\}_{i=4}^{14}$ can be proved in zero-knowledge using random permutations.

Meanwhile, we need a more sophisticated treatment for the components of vector $\mathbf{w}_{15}$. For $j = 1, 2$, we let $\mathbf{d}''_j = \mathsf{ExtendThree}_{m\delta_\beta}(\mathbf{d}'_j) \in \mathsf{B}_{3m\delta_\beta}$. We also extend τ to $\tau'' = \mathsf{ExtendTwo}_\ell(\tau) = (\tau[1], \ldots, \tau[\ell], \tau[\ell+1], \ldots, \tau[2\ell])^\top \in \mathsf{B}_{2\ell}$. Then we form the vector:

$$\mathbf{w}''_{15} = \big((\mathbf{d}''_1)^\top \| (\mathbf{d}''_2)^\top \| \tau[1](\mathbf{d}''_2)^\top \| \ldots \| \tau[\ell](\mathbf{d}''_2)^\top \| \ldots \| \tau[2\ell](\mathbf{d}''_2)^\top \big)^\top.$$

Next, we define CorMix as the set of all vectors in $\{-1, 0, 1\}^{(2\ell+2)3m\delta_\beta}$, that have the form $\big(\mathbf{z}_1^\top \| \mathbf{z}_2^\top \| \rho[1]\mathbf{z}_2^\top \| \ldots \| \rho[2\ell]\mathbf{z}_2^\top \big)^\top$ for some $\mathbf{z}_1, \mathbf{z}_2 \in \mathsf{B}_{3m\delta_\beta}$ and $\rho \in \mathsf{B}_{2\ell}$. Clearly, $\mathbf{w}''_{15} \in \mathsf{CorMix}$. Furthermore, as shown in [42,38], this set is closed under a special composition of 3 permutations $\phi_1 \in \mathcal{S}_{3m\delta_\beta}, \phi_2 \in \mathcal{S}_{3m\delta_\beta}, \phi_3 \in \mathcal{S}_{2\ell}$, which we denote by T_{ϕ_1,ϕ_2,ϕ_3}. That is, we have the equivalence:

$$\mathbf{w}''_{15} \in \mathsf{CorMix} \iff T_{\phi_1,\phi_2,\phi_3}(\mathbf{w}''_{15}) \in \mathsf{CorMix}. \tag{10}$$

– As we have changed the dimensions of the witness-vectors, we also have to transform the public matrices $\{\mathbf{M}_{i,j}\}_{i,j}$ accordingly to preserve the equations. In short, this can be done through right-multiplying by the decomposition matrices (if needed), and then inserting zero-columns at suitable positions. We denote the transformed public matrices by $\{\mathbf{M}''_{i,j}\}_{i,j}$.

At the end of the second step, we are presented with the following system of equations, which is equivalent to (9).

$$\begin{cases} \mathbf{v}_1 = \mathbf{M}''_{1,1} \cdot \mathbf{w}''_1 + \mathbf{M}''_{1,2} \cdot \mathbf{w}''_2 + \ldots + \mathbf{M}''_{1,15} \cdot \mathbf{w}''_{15} \bmod q, \\ \mathbf{v}_2 = \mathbf{M}''_{2,1} \cdot \mathbf{w}''_1 + \mathbf{M}''_{2,2} \cdot \mathbf{w}''_2 + \ldots + \mathbf{M}''_{2,15} \cdot \mathbf{w}''_{15} \bmod q, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mathbf{v}_{10} = \mathbf{M}''_{10,1} \cdot \mathbf{w}''_1 + \mathbf{M}''_{10,2} \cdot \mathbf{w}''_2 + \ldots + \mathbf{M}''_{10,15} \cdot \mathbf{w}''_{15} \bmod q. \end{cases} \tag{11}$$

The third step involves only basic linear algebra. Let

$$\mathbf{M} = \left(\begin{array}{c|c|c|c} \mathbf{M}''_{1,1} & \mathbf{M}''_{1,2} & \ldots & \mathbf{M}''_{1,15} \\ \hline \mathbf{M}''_{2,1} & \mathbf{M}''_{2,2} & \ldots & \mathbf{M}''_{2,15} \\ \hline \ldots & \ldots & \ldots & \ldots \\ \hline \mathbf{M}''_{10,1} & \mathbf{M}''_{10,2} & \ldots & \mathbf{M}''_{10,15} \end{array}\right); \quad \mathbf{x} = \begin{pmatrix} \mathbf{w}''_1 \\ \mathbf{w}''_2 \\ \vdots \\ \mathbf{w}''_{15} \end{pmatrix}; \quad \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{10} \end{pmatrix},$$

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

Given the above preparations, we now comes to **the final step** where we formally present our protocol. Let D be the dimension of vector $\mathbf{x}$. Denote by VALID the set of all vectors in $\{-1, 0, 1\}^D$, that have the form $\mathbf{z} = (\mathbf{z}_1^\top \| \ldots \| \mathbf{z}_{15}^\top)^\top$, where:

1. $\mathbf{z}_1 = \mathsf{encode}_{n\bar{m}k}(\mathbf{y}_1)$, $\mathbf{z}_2 = \mathsf{encode}_{nk}(\mathbf{y}_2)$ and $\mathbf{z}_3 = \mathsf{expand}^\otimes(\mathbf{y}_1, \mathbf{y}_2)$, for some $\mathbf{y}_1 \in \{0,1\}^{n\bar{m}k}$ and $\mathbf{y}_2 \in \{0,1\}^{nk}$.
2. For $i \in [4, 7]$, vector $\mathbf{z}_i \in \mathsf{B}_{2m_i}$. For $i \in [8, 14]$, vector $\mathbf{z}_i \in \mathsf{B}_{3m_i\delta_{\beta_i}}$.

3. Vector $\mathbf{z}_{15} \in \mathsf{CorMix}$.

It can be seen that our vector $\mathbf{x}$ is an element of this tailored set VALID. By construction, the task of proving knowledge of vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{15}$ that have the required constraints, and that satisfy system (9) has boiled down to proving the possession of vector $\mathbf{x} \in \mathsf{VALID}$ such that $\mathbf{M} \cdot \mathbf{x} = \mathbf{v} \bmod q$. We will fulfill this task with a Stern-like zero-knowledge protocol, in which we hide $\mathbf{x}$ from the verifier's view by a random permutation and a random masking vector.

Let us determine the type of permutations to be applied for $\mathbf{x}$. Let

$$\mathcal{S} = \{0,1\}^{n\bar{m}k} \times \{0,1\}^{nk} \times \mathcal{S}_{2m_4} \times \ldots \times \mathcal{S}_{2m_7} \times \mathcal{S}_{3m_8\delta_{\beta_8}} \times \ldots \\ \ldots \times \mathcal{S}_{3m_{14}\delta_{\beta_{14}}} \times (\mathcal{S}_{3m\delta_\beta})^2 \times \mathcal{S}_{2\ell}.$$

We associate each element $\pi = (\mathbf{b}_1, \mathbf{b}_2, \phi_4, \ldots, \phi_{14}, \phi_{15}^1, \phi_{15}^2, \phi_{15}^3) \in \mathcal{S}$ with the permutation Γ_π that transforms vector $\mathbf{z} = (\mathbf{z}_1^\top \| \ldots \| \mathbf{z}_{15}^\top)^\top \in \mathbb{Z}^D$, where the length of block $\mathbf{z}_i$ equals to that of $\mathbf{w}_i''$ for all $i \in [15]$, into vector

$$\Gamma_\pi(\mathbf{z}) = \big(F_{\mathbf{b}_1}^{(n\bar{m}k)}(\mathbf{z}_1) \| F_{\mathbf{b}_2}^{(nk)}(\mathbf{z}_2) \| P_{\mathbf{b}_1,\mathbf{b}_2}(\mathbf{z}_3) \| \phi_4(\mathbf{z}_4) \| \ldots \\ \ldots \| \phi_{14}(\mathbf{z}_{14}) \| T_{\phi_{15}^1,\phi_{15}^2,\phi_{15}^3}(\mathbf{z}_{15})\big).$$

It is implied by the equivalences given in (3), (5), (4) and (10) that the following holds for all $\pi \in \mathcal{S}$:

$$\mathbf{x} \in \mathsf{VALID} \iff \Gamma_\pi(\mathbf{x}) \in \mathsf{VALID}.$$

Additionally, if $\mathbf{x} \in \mathsf{VALID}$ and π is uniformly random in $\mathcal{S}$, then $\Gamma_\pi(\mathbf{x})$ is uniformly random in VALID. In the framework of Stern's protocol, these facts allow us to prove in zero-knowledge the knowledge of $\mathbf{x} \in \mathsf{VALID}$.

Furthermore, proving that equation $\mathbf{M} \cdot \mathbf{x} = \mathbf{v} \bmod q$ holds can be done by sampling a uniformly random masking vector $\mathbf{r}_x \in \mathbb{Z}_q^D$, and demonstrating to the verifier that $\mathbf{M} \cdot (\mathbf{x} + \mathbf{r}_x) - \mathbf{v} = \mathbf{M} \cdot \mathbf{r}_x \bmod q$.

The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 1. Prior to the interaction, both parties obtain matrix $\mathbf{M}$ and vector $\mathbf{v}$ from the public input, while $\mathcal{P}$ construct witness-vector $\mathbf{x}$ from his secret input, as described above. The protocol employs the statistically hiding and computationally binding string commitment scheme COM from [32].

The properties of the given protocol are summarized in Theorem 1. The proof of the theorem employs standard simulation and extraction techniques for Stern-like protocols [32,41,42], and is deferred to Appendix B.

Theorem 1. *The protocol in Figure 1 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(D \log q)$*. Namely:*

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

1. **Commitment:** Prover samples $\mathbf{r}_x \leftarrow U(\mathbb{Z}_q^D)$, $\pi \leftarrow U(\mathcal{S})$ 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

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

2. **Challenge:** The verifier sends a challenge $Ch \leftarrow U(\{1, 2, 3\})$ to the prover.
3. **Response:** Depending on Ch, the prover sends RSP computed as follows:
   - $Ch = 1$: Let $\mathbf{t}_x = \Gamma_\pi(\mathbf{x})$, $\mathbf{t}_r = \Gamma_\pi(\mathbf{r}_x)$, and $\mathrm{RSP} = (\mathbf{t}_x, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\pi_2 = \pi$, $\mathbf{y}_2 = \mathbf{x} + \mathbf{r}_x$, and $\mathrm{RSP} = (\pi_2, \mathbf{y}_2, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\pi_3 = \pi$, $\mathbf{y}_3 = \mathbf{r}$, and $\mathrm{RSP} = (\pi_3, \mathbf{y}_3, \rho_1, \rho_2)$.

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 $C_1 = \mathsf{COM}(\pi_2, \mathbf{M} \cdot \mathbf{y}_2 - \mathbf{v}; \rho_1)$, $C_3 = \mathsf{COM}(\Gamma_{\pi_2}(\mathbf{y}_2); \rho_3)$.
- $Ch = 3$: Check that $C_1 = \mathsf{COM}(\pi_3, \mathbf{M} \cdot \mathbf{y}_3; \rho_1)$, $C_2 = \mathsf{COM}(\Gamma_{\pi_3}(\mathbf{y}_3); \rho_2)$.

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

Fig. 1: Our zero-knowledge argument of knowledge.

- *There exists a polynomial-time 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* $\mathbf{M} \cdot \mathbf{x}' = \mathbf{v} \bmod q$.

Note that, given vector $\mathbf{x}'$ outputted by the extractor, one can efficiently compute 15 vectors satisfying the conditions described at the beginning of this subsection, simply by "backtracking" the transformations conducted by our first 3 steps. In the group encryption scheme presented next, the constructed ZKAoK will be invoked by algorithm $\langle \mathcal{P}, \mathcal{V} \rangle$, while its simulator and extractor will come in handy in the proofs of Theorems 2, 3, and 4.

5 Our Lattice-Based Group Encryption Scheme

To build a GE scheme using our zero-knowledge argument system, we need to choose a specific key-private CCA2-secure encryption scheme. The first idea is to use the CCA2-secure public-key cryptosystem which is implied by the Agrawal-Boneh-Boyen identity-based encryption (IBE) scheme [1] (which is recalled in Appendix A.2) via the Canetti-Halevi-Katz (CHK) transformation [16]. The ABB scheme is a natural choice since it has pseudo-random ciphertexts (which implies the key-privacy [7] when the CHK paradigm is applied) and provides one of the most efficient CCA2 cryptosystem based on the hardness of LWE in the standard model. One difficulty is that the Kiayias-Tsiounis-Yung model [34] requires that certified public keys be valid public keys (i.e., which have a matching secret key). When new group members join the system and request a certificate for their public key $\mathbf{B}_\mathsf{U} \in \mathbb{Z}_q^{n \times \bar{m}}$, a direct use of the ABB/CHK technique would incur of proof of existence of a GPV trapdoor [24] corresponding to $\mathbf{B}_\mathsf{U}$ (i.e., a

small-norm matrix $\mathbf{T}_{\mathbf{B}_\mathsf{U}} \in \mathbb{Z}^{\bar{m}\times\bar{m}}$ s.t. $\mathbf{B}\cdot\mathbf{T}_{\mathbf{B}_\mathsf{U}} = \mathbf{0}^n \bmod q$). While the techniques of Peikert and Vaikuntanathan [49] would provide a solution to this problem (as they allow proving that $\mathbf{T}_{\mathbf{B}_\mathsf{U}} \in \mathbb{Z}^{\bar{m}\times\bar{m}}$ has full-rank), we found it simpler to rely on the trapdoor mechanism of Micciancio and Peikert [44].

If we assume public parameters containing a random matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$, each user's public key can consist of a matrix $\mathbf{B}_\mathsf{U} = \bar{\mathbf{A}}\cdot\mathbf{T}_\mathsf{U} \in \mathbb{Z}_q^{n\times\bar{m}}$, where $\mathbf{T}_\mathsf{U} \in \mathbb{Z}^{m\times\bar{m}}$ is a small-norm matrix whose calms are sampled from a discrete Gaussian distribution. Note that, if $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ is uniformly distributed, then [24, Lemma 5.1] ensures that, with overwhelming probability, any matrix $\mathbf{B}_\mathsf{U} \in \mathbb{Z}_q^{n\times\bar{m}}$ has an underlying small-norm matrix satisfying $\mathbf{B}_\mathsf{U} = \bar{\mathbf{A}}\cdot\mathbf{T}_\mathsf{U} \bmod q$. This simplifies the joining procedure by eliminating the need for proofs of public key validity.

In the encryption algorithm, the sender computes a dual Regev encryption [24] of the witness $\mathbf{w} \in \{0,1\}^m$ using a matrix $[\bar{\mathbf{A}} \mid \mathbf{B}_\mathsf{U} + \mathsf{FRD}(\mathsf{VK})\cdot\mathbf{G}] \in \mathbb{Z}_q^{n\times(m+\bar{m})}$ such that: (i) $\mathsf{VK} \in \mathbb{Z}_q^n$ is the verification key of a one-time signature; (ii) $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is the full-rank difference[3] function of [1]; (iii) $\mathbf{G} = \mathbf{I}_n \otimes [1|2|\ldots|2^{k-1}] \in \mathbb{Z}_q^{n\times\bar{m}}$ is the gadget matrix of [44]. Given that $\mathbf{G}$ has a publicly known trapdoor allowing to sample short vectors in $\Lambda_q^\perp(\mathbf{G})$, the user can use his private key $\mathbf{T}_\mathsf{U} \in \mathbb{Z}^{m\times\bar{m}}$ to decrypt by running the $\mathsf{SampleRight}$ algorithm of Lemma 5.

Having encrypted the witness $\mathbf{w} \in \{0,1\}^m$ by running the ABB encryption algorithm, the sender proceeds by encrypting a hash value of $\mathbf{B}_\mathsf{U} \in \mathbb{Z}_q^{n\times\bar{m}}$ under the public key $\mathbf{B}_\mathsf{OA} = \bar{\mathbf{A}}\cdot\mathbf{T}_\mathsf{OA} \in \mathbb{Z}_q^{n\times\bar{m}}$ of the opening authority. The latter hash value is obtained as a bit-wise decomposition of $\mathbf{F}\cdot\mathsf{mdec}_{n,m,q}(\mathbf{B}_\mathsf{U}^\top) \in \mathbb{Z}_q^{2n}$, where $\mathbf{F} \in \mathbb{Z}_q^{2n\times n\bar{m}\lceil\log q\rceil}$ is a random public matrix and $\mathsf{mdec}_{n,m,q}(\mathbf{B}_\mathsf{U}^\top) \in \{0,1\}^{n\bar{m}\lceil\log q\rceil}$ denotes an entry-wise binary decomposition of the matrix $\mathbf{B}_\mathsf{U} \in \mathbb{Z}_q^{n\times\bar{m}}$.

By combining our new argument for quadratic relations and the extensions of Stern's protocol suggested in [42,38], we are able to prove that some component of the ciphertext is of the form $\mathbf{c} = \mathbf{B}_\mathsf{U}^\top\cdot\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^{\bar{m}}$, for some $\mathbf{s} \in \mathbb{Z}_q^n$ and a small-norm $\mathbf{e} \in \mathbb{Z}^{\bar{m}}$ while also arguing possession of a signature on the binary decomposition $\mathsf{mdec}_{n,m,q}(\mathbf{B}_\mathsf{U}^\top) \in \{0,1\}^{n\bar{m}\lceil\log q\rceil}$ of $\mathbf{B}_\mathsf{U}^\top$. For this purpose, we use a variant of a signature scheme due to Böhl *et al.*'s signature [11] which was recently proposed by Libert, Ling, Mouhartem, Nguyen and Wang [38] (and of which a description is given in Appendix A.1). At the same time, the prover $\mathcal{P}$ can also argue that a hash value of $\mathsf{mdec}_{n,m,q}(\mathbf{B}_\mathsf{U}^\top)$ is properly encrypted under the OA's public key using the ABB encryption scheme.

5.1 Description of the Scheme

Our GE scheme allows encrypting witnesses for the Inhomogeneous SIS relation $\mathrm{R}_{\mathsf{ISIS}}(n,m,q,1)$, which consists of pairs $((\mathbf{A}_R,\mathbf{u}_R),\mathbf{w}) \in (\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^n)\times\{0,1\}^m$ satisfying $\mathbf{u}_R = \mathbf{A}_R\cdot\mathbf{w} \bmod q$. This relation is in the same spirit as the one of Kiayias, Tsiounis and Yung [34], who consider the verifiable encryption of

[3] This means that, for any two distinct one-time verification keys $\mathsf{VK},\mathsf{VK}' \in \mathbb{Z}_q^n$, the difference $\mathsf{FRD}(\mathsf{VK}) - \mathsf{FRD}(\mathsf{VK}') \in \mathbb{Z}_q^{n\times n}$ is invertible over $\mathbb{Z}_q$.

discrete logarithms. While the construction of [34] allow verifiably encrypting discrete-logarithm-type secret keys under the public key of some anonymous TTP, our construction makes it possible to encrypt GPV-type secret keys [24].

$\mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$**:** This algorithm performs the following:

1. Choose integers $n = \mathcal{O}(\lambda)$, prime $q = \widetilde{\mathcal{O}}(n^4)$, and let $k = \lceil \log_2 q \rceil$, $\bar{m} = nk$ and $m = 2\bar{m} = 2nk$. Choose a B-bounded distribution χ over $\mathbb{Z}$ for some $B = \sqrt{n}\omega(\log n)$.
2. Choose a Gaussian parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$. Let $\beta = \sigma\omega(\log n)$ be the upper bound of samples from $D_{\mathbb{Z},\sigma}$.
3. Select integers $\ell = \ell(\lambda)$ which determines the maximum expected group size 2^ℓ, and $\kappa = \omega(\log \lambda)$ (the number of protocol repetitions).
4. Select a strongly unforgeable one-time signature $\mathcal{OTS} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$. We assume that the verification keys live in $\mathbb{Z}_q^n$.
5. Select public parameters $\mathsf{COM}_{\mathsf{par}}$ for a statistically-hiding commitment scheme like [32]. This commitment will serve as a building block for the zero-knowledge argument system used in $\langle \mathcal{P}, \mathcal{V} \rangle$.
6. Let $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ be the full-rank difference mapping from [1].
7. Pick a random matrix $\mathbf{F} \leftarrow \mathbb{Z}_q^{2n \times n\bar{m}k}$, which will be used to hash users' public keys from $\mathbb{Z}_q^{n \times \bar{m}}$ to $\mathbb{Z}_q^n$.
8. Let $\mathbf{G} \in \mathbb{Z}_q^{n \times \bar{m}}$ be the gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes \left[1\ 2 \dots 2^{k-1}\right]$ of [44]. Pick matrices $\bar{\mathbf{A}}, \mathbf{U} \leftarrow U(\mathbb{Z}_q^{n \times m})$ and $\mathbf{V} \leftarrow U(\mathbb{Z}_q^{n \times m})$. Looking ahead, $\mathbf{U}$ will be used to encrypt for the receiver while $\mathbf{V}$ will be used to encrypt the user's public key under the OA's public key. As for $\bar{\mathbf{A}}$, it will be used in two instances of the ABB encryption scheme [1].

Output

$$\mathsf{param} = \big\{\lambda, n, q, k, m, B, \chi, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\big\}.$$

$\langle \mathcal{G}_r, \mathsf{sample}_{\mathcal{R}} \rangle$**:** Algorithm $\mathcal{G}_r(1^\lambda, 1^n, 1^m)$ proceeds by sampling a random matrix $\mathbf{A}_R \leftarrow U(\mathbb{Z}_q^{n \times m})$ and outputting $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}) = (\mathbf{A}_R, \varepsilon)$. On input of a public key $\mathsf{pk}_{\mathcal{R}} = \mathbf{A}_R \in \mathbb{Z}_q^{n \times m}$ for the relation $\mathrm{R}_{\mathsf{ISIS}}$, algorithm $\mathsf{sample}_{\mathcal{R}}$ picks $\mathbf{w} \leftarrow U(\{0,1\}^m)$ and outputs a pair $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w})$, where $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \in \mathbb{Z}_q^n$.

$\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$**:** The GM generates $(\mathsf{sk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{GM}}) \leftarrow \mathsf{Keygen}(1^\lambda, q, n, m, \ell, \sigma)$ as a key pair for the SIS-based signature scheme of [38] (as recalled in Appendix A.1). This key pair consists of $\mathsf{sk}_{\mathsf{GM}} := \mathbf{T}_{\mathbf{A}}$ and

$$\mathsf{pk}_{\mathsf{GM}} := \Big(\mathbf{A}, \mathbf{A}_0, \dots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n \times m},\ \mathbf{D}_0, \mathbf{D}_1 \in \mathbb{Z}_q^{n \times m}, \mathbf{D} \in \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{u} \in \mathbb{Z}_q^n\Big). \quad (12)$$

$\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$**:** The OA samples a small-norm matrix $\mathbf{T}_{\mathsf{OA}} \leftarrow D_{\mathbb{Z}^{\bar{m}},\sigma}^{\bar{m}}$ in $\mathbb{Z}^{m \times \bar{m}}$ to obtain a statistically uniform $\mathbf{B}_{\mathsf{OA}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$. The OA's key pair consists of $(\mathsf{sk}_{\mathsf{OA}}, \mathsf{pk}_{\mathsf{OA}}) = (\mathbf{T}_{\mathsf{OA}}, \mathbf{B}_{\mathsf{OA}})$.

JOIN**:** The prospective user U and the GM interact in the following protocol.

1. U first samples $\mathbf{T}_{\mathsf{U}} \leftarrow D^{\bar{m}}_{\mathbb{Z}^m,\sigma}$ in $\mathbb{Z}^{m\times\bar{m}}$ to compute a statistically uniform matrix $\mathbf{B}_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$. The prospective user defines his key pair as $(\mathsf{pk}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}}) = (\mathbf{B}_{\mathsf{U}}, \mathbf{T}_{\mathsf{U}})$ and sends $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$ to the GM.
2. Upon receiving a public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ from the user, the GM certifies pk_U via the following steps:
   a. Compute $\mathbf{h}_{\mathsf{U}} = \mathbf{F}\cdot\mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^{\top}) \in \mathbb{Z}_q^{2n}$ as a hash value of the public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$.
   b. Use the trapdoor $\mathsf{sk}_{\mathsf{GM}} = \mathbf{T}_{\mathbf{A}}$ to generate a signature
   $$\mathsf{cert}_{\mathsf{U}} = \big(\tau, \mathbf{d}, \mathbf{r}\big) \in \{0,1\}^{\ell} \times [-\beta,\beta]^{2m} \times [-\beta,\beta]^{m}, \tag{13}$$
   satisfying
   $$\big[\mathbf{A} \mid \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j\big] \cdot \mathbf{d} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U}})) \bmod q, \tag{14}$$
   where $\tau = \tau[1]\ldots\tau[\ell] \in \{0,1\}^{\ell}$, as in the scheme of Section A.1.

   U verifies that $\mathsf{cert}_{\mathsf{U}}$ is tuple of the form (13) satisfying (14) and returns $\perp$ if it is not the case. The GM stores $(\mathsf{pk}_{\mathsf{U}}, \mathsf{cert}_{\mathsf{U}})$ in the user database $\mathsf{database}$ and returns the certificate $\mathsf{cert}_{\mathsf{U}}$ to the new user $\mathcal{U}$.

$\mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathsf{U}}, \mathsf{cert}_{\mathsf{U}}, \mathbf{w}, L)$**:** To encrypt a witness $\mathbf{w} \in \{0,1\}^m$ such that $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}) \in \mathrm{R}_{\mathsf{ISIS}}(n, m, q, 1)$ (i.e., $\mathbf{A}_R \cdot \mathbf{w} = \mathbf{u}_R \bmod q$), parse $\mathsf{pk}_{\mathsf{GM}}$ as in (12), $\mathsf{pk}_{\mathsf{OA}}$ as $\mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$, pk_{U} as $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathsf{cert}_{\mathsf{U}}$ as in (13).

1. Generate a one-time key-pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathsf{Gen}(1^{\lambda})$, where $\mathsf{VK} \in \mathbb{Z}_q^n$.
2. Compute a full-rank-difference hash $\mathbf{H}_{\mathsf{VK}} = \mathsf{FRD}(\mathsf{VK}) \in \mathbb{Z}_q^{n\times n}$ of the one-time verification key $\mathsf{VK} \in \mathbb{Z}_q^n$.
3. Encrypt the witness $\mathbf{w} \in \{0,1\}^m$ under U's public key $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ using the tag VK by taking the following steps:
   a. Sample $\mathbf{s}_{\mathsf{rec}} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{R}_{\mathsf{rec}} \leftarrow D^{m\times\bar{m}}_{\mathbb{Z},\sigma}$ and $\mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}} \leftarrow \chi^m$. Compute $\mathbf{z}_{\mathsf{rec}} = \mathbf{R}_{\mathsf{rec}}^{\top} \cdot \mathbf{y}_{\mathsf{rec}} \in \mathbb{Z}^{\bar{m}}$.
   b. Compute
   $$\begin{cases} \mathbf{c}_{\mathsf{rec}}^{(1)} = \bar{\mathbf{A}}^{\top} \cdot \mathbf{s}_{\mathsf{rec}} + \mathbf{y}_{\mathsf{rec}} \bmod q \\ \mathbf{c}_{\mathsf{rec}}^{(2)} = (\mathbf{B}_{\mathsf{U}} + \mathbf{H}_{\mathsf{VK}} \cdot \mathbf{G})^{\top} \cdot \mathbf{s}_{\mathsf{rec}} + \mathbf{z}_{\mathsf{rec}} \bmod q; \\ \mathbf{c}_{\mathsf{rec}}^{(3)} = \mathbf{U}^{\top} \cdot \mathbf{s}_{\mathsf{rec}} + \mathbf{x}_{\mathsf{rec}} + \mathbf{w} \cdot \left\lfloor \frac{q}{2} \right\rfloor, \end{cases} \tag{15}$$
   and let $\mathbf{c}_{\mathsf{rec}} = \big(\mathbf{c}_{\mathsf{rec}}^{(1)}, \mathbf{c}_{\mathsf{rec}}^{(2)}, \mathbf{c}_{\mathsf{rec}}^{(3)}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$, which forms an ABB ciphertext [1] for the tag $\mathsf{VK} \in \mathbb{Z}_q^n$.
4. Encrypt the decomposition $\mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U}}) \in \{0,1\}^m$ of the hashed pk_{U} under the OA's public key $\mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$ w.r.t. the tag $\mathsf{VK} \in \mathbb{Z}_q^n$. Namely, conduct the following steps:

a. Sample $\mathbf{s}_{\mathsf{oa}} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{R}_{\mathsf{oa}} \leftarrow D_{\mathbb{Z},\sigma}^{m\times\bar{m}}$, $\mathbf{x}_{\mathsf{oa}} \leftarrow \chi^m, \mathbf{y}_{\mathsf{oa}} \leftarrow \chi^m$. Set $\mathbf{z}_{\mathsf{oa}} = \mathbf{R}_{\mathsf{oa}}^\top \cdot \mathbf{y}_{\mathsf{oa}} \in \mathbb{Z}^{\bar{m}}$.

b. Compute

$$\begin{cases} \mathbf{c}_{\mathsf{oa}}^{(1)} = \bar{\mathbf{A}}^\top \cdot \mathbf{s}_{\mathsf{oa}} + \mathbf{y}_{\mathsf{oa}} \bmod q; \\ \mathbf{c}_{\mathsf{oa}}^{(2)} = (\mathbf{B}_{\mathsf{OA}} + \mathbf{H}_{\mathsf{VK}} \cdot \mathbf{G})^\top \cdot \mathbf{s}_{\mathsf{oa}} + \mathbf{z}_{\mathsf{oa}} \bmod q; \\ \mathbf{c}_{\mathsf{oa}}^{(3)} = \mathbf{V}^\top \cdot \mathbf{s}_{\mathsf{oa}} + \mathbf{x}_{\mathsf{oa}} + \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U}}) \cdot \left\lfloor \frac{q}{2} \right\rfloor, \end{cases} \tag{16}$$

and let $\mathbf{c}_{\mathsf{oa}} = \big(\mathbf{c}_{\mathsf{oa}}^{(1)}, \mathbf{c}_{\mathsf{oa}}^{(2)}, \mathbf{c}_{\mathsf{oa}}^{(3)}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$.

5. Compute a one-time signature $\Sigma = \mathsf{Sig}(\mathsf{SK}, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, L))$.

Output the ciphertext

$$\boldsymbol{\Psi} = (\mathsf{VK}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, \Sigma). \tag{17}$$

and the state information $coins_{\boldsymbol{\Psi}} = \big(\mathbf{s}_{\mathsf{rec}}, \mathbf{R}_{\mathsf{rec}}, \mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}}, \mathbf{s}_{\mathsf{oa}}, \mathbf{R}_{\mathsf{oa}}, \mathbf{x}_{\mathsf{oa}}, \mathbf{y}_{\mathsf{oa}}\big)$.

$\mathsf{DEC}(\mathsf{sk}_{\mathsf{U}}, \boldsymbol{\Psi}, L)$: The decryption algorithm proceeds as follows:

1. If $\mathsf{Ver}\big(\mathsf{VK}, \Sigma, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, L)\big) = 0$, return $\perp$. Otherwise, parse the secret key sk_{U} as $\mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m\times\bar{m}}$ and the ciphertext $\boldsymbol{\Psi}$ as in (17). Define the matrix $\mathbf{B}_{\mathsf{VK}} = \mathbf{B}_{\mathsf{U}} + \mathsf{FRD}(\mathsf{VK}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n\times\bar{m}}$.
2. Decrypt $\mathbf{c}_{\mathsf{rec}}$ using a decryption key for the tag $\mathsf{VK} \in \mathbb{Z}^n$. Namely,
   a. Define $\mathbf{B}_{\mathsf{U},\mathsf{VK}} = [\bar{\mathbf{A}}|\mathbf{B}_{\mathsf{VK}}] = [\bar{\mathbf{A}}|\bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} + \mathsf{FRD}(\mathsf{VK}) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n\times(m+\bar{m})}$. Using $\mathbf{T}_{\mathsf{U}}$ and the publicly known trapdoor $\mathbf{T}_{\mathbf{G}}$ of $\mathbf{G}$, compute a small-norm matrix $\mathbf{E}_{\mathsf{VK}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ such that $\mathbf{B}_{\mathsf{U},\mathsf{VK}} \cdot \mathbf{E}_{\mathsf{VK}} = \mathbf{U} \bmod q$ by running the $\mathsf{SampleRight}$ algorithm of Lemma 5.
   b. Compute

$$\mathbf{w} = \left\lfloor \left(\mathbf{c}_{\mathsf{rec}}^{(3)} - \mathbf{E}_{\mathsf{VK}}^\top \cdot \begin{bmatrix} \mathbf{c}_{\mathsf{rec}}^{(1)} \\ \mathbf{c}_{\mathsf{rec}}^{(2)} \end{bmatrix}\right) / \left\lfloor \frac{q}{2} \right\rfloor \right\rceil \in \mathbb{Z}^m$$

   and return the obtained $\mathbf{w} \in \{0,1\}^m$.

$\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \boldsymbol{\Psi}, L)$: The opening algorithm proceeds as follows:

1. If $\mathsf{Ver}\big(\mathsf{VK}, \Sigma, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}), L\big) = 0$, then return $\perp$. Otherwise, parse $\mathsf{sk}_{\mathsf{OA}}$ as $\mathbf{T}_{\mathsf{OA}} \in \mathbb{Z}^{m\times\bar{m}}$ and the ciphertext $\boldsymbol{\Psi}$ as in (17).
2. Decrypt $\mathbf{c}_{\mathsf{oa}}$ using a decryption key for the tag $\mathsf{VK} \in \mathbb{Z}_q^n$ in the same way as in the decryption algorithm. That is, do the following:
   a. Define the matrix $\mathbf{B}_{\mathsf{OA},\mathsf{VK}} = [\bar{\mathbf{A}}|\mathbf{B}_{\mathsf{OA}} + \mathsf{FRD}(\mathsf{VK}) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n\times(m+\bar{m})}$. Use $\mathbf{T}_{\mathsf{OA}}$ to compute a small-norm $\mathbf{E}_{\mathsf{OA},\mathsf{VK}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ satisfying $\mathbf{B}_{\mathsf{OA},\mathsf{VK}} \cdot \mathbf{E}_{\mathsf{OA},\mathsf{VK}} = \mathbf{V} \bmod q$.
   b. Compute

$$\mathbf{h} = \left\lfloor \left(\mathbf{c}_{\mathsf{oa}}^{(3)} - \mathbf{E}_{\mathsf{OA},\mathsf{VK}}^\top \cdot \begin{bmatrix} \mathbf{c}_{\mathsf{oa}}^{(1)} \\ \mathbf{c}_{\mathsf{oa}}^{(2)} \end{bmatrix}\right) / \left\lfloor \frac{q}{2} \right\rfloor \right\rceil \in \{0,1\}^m$$

   and $\mathbf{h}'_{\mathsf{U}} = \mathbf{H}_{2n,q-1} \cdot \mathbf{h} \in \mathbb{Z}_q^{2n}$.

3. Look up $\mathsf{database}$ to find a public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ that hashes to $\mathbf{h}'_{\mathsf{U}} \in \mathbb{Z}_q^{2n}$ (i.e., such that $\mathbf{h}'_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^{\top})$). If more than one such key exists, return $\perp$. If only one key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ satisfies $\mathbf{h}'_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^{\top})$, return that key pk_{U}. In any other situation, return $\perp$.

$\langle \mathcal{P}, \mathcal{V} \rangle$**:** The common input consists of param and $\mathsf{pk}_{\mathsf{GM}}$ as specified above, as well as $(\mathbf{A}_R, \mathbf{u}_R) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$, $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$, and a ciphertext $\boldsymbol{\Psi}$ as in (17). Both parties compute $\mathbf{B}_{\mathsf{OA},\mathsf{VK}} = [\bar{\mathbf{A}} | \mathbf{B}_{\mathsf{OA}} + \mathsf{FRD}(\mathsf{VK}) \cdot \mathbf{G}]$ as specified above. The prover's secret input consists of a witness $\mathbf{w} \in \{0,1\}^m$, $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$, $\mathsf{cert}_{\mathsf{U}} = (\tau, \mathbf{d}, \mathbf{r}) \in \{0,1\}^{\ell} \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$, and the random coins $coins_{\boldsymbol{\Psi}} = \big(\mathbf{s}_{\mathsf{rec}}, \mathbf{R}_{\mathsf{rec}}, \mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}}, \mathbf{s}_{\mathsf{oa}}, \mathbf{R}_{\mathsf{oa}}, \mathbf{x}_{\mathsf{oa}}, \mathbf{y}_{\mathsf{oa}}\big)$ used to generate $\boldsymbol{\Psi}$.

The prover's goal is to convince the verifier in zero-knowledge that his secret input satisfies the following:

1. $\mathbf{A}_R \cdot \mathbf{w} = \mathbf{u}_R \bmod q$.
2. $\mathbf{h}_{\mathbf{M}} = \mathbf{F} \cdot \mathsf{mdec}_{n,m,q}(\mathbf{M}) \bmod q$.
3. Conditions (13) and (14) hold.
4. Vectors $\mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}}, \mathbf{x}_{\mathsf{oa}}, \mathbf{y}_{\mathsf{oa}}$ have infinity norms bounded by B, and vectors $\mathbf{z}_{\mathsf{rec}}, \mathbf{z}_{\mathsf{oa}}$ have infinity norms bounded by $\beta m B$.
5. Equations in (15) and (16) hold.

To this end $\mathcal{P}$ conducts the following steps.

1. Decompose the matrix $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ into $\mathbf{b}_{\mathsf{U}} = \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^{\top}) \in \{0,1\}^{n\bar{m}k}$ and the vectors $\mathbf{s}_{\mathsf{rec}}, \mathbf{s}_{\mathsf{oa}} \in \mathbb{Z}_q^n$ into $\mathbf{s}_{0,\mathsf{rec}} = \mathsf{vdec}_{n,q-1}(\mathbf{s}_{\mathsf{rec}}) \in \{0,1\}^{nk}$ and $\mathbf{s}_{0,\mathsf{oa}} = \mathsf{vdec}_{n,q-1}(\mathbf{s}_{\mathsf{oa}}) \in \{0,1\}^{nk}$. Combine the first two binary vectors into $\mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}}) \in \{0,1\}^{4n\bar{m}k^2}$. Define

$$\mathbf{Q} = \mathbf{H}_{\bar{m},q-1} \cdot [\overbrace{\mathbf{Q}_0 | \ldots | \mathbf{Q}_0}^{n \text{ times}}] \in \mathbb{Z}_q^{\bar{m} \times 4n\bar{m}k^2},$$

where $\mathbf{Q}_0 = \mathbf{I}_{\bar{m}k} \otimes \mathbf{g}' \in \mathbb{Z}_q^{\bar{m}k \times 4\bar{m}k^2}$ is the matrix defined as in (7).

2. Generate a zero-knowledge argument of knowledge of

$$\begin{cases} \tau \in \{0,1\}^{\ell},\ \mathbf{d} = [\mathbf{d}_1^{\top} | \mathbf{d}_2^{\top}]^{\top} \in [-\beta, \beta]^{2m},\ \mathbf{r} \in [-\beta, \beta]^m \\ \mathbf{t}_{\mathsf{U}} \in \{0,1\}^m,\ \mathbf{w}_{\mathsf{U}} \in \{0,1\}^{\bar{m}} \\ \mathbf{b}_{\mathsf{U}} \in \{0,1\}^{n\bar{m}k},\ \mathbf{s}_{0,\mathsf{rec}} \in \{0,1\}^{nk},\ \mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}}) \\ \mathbf{x}_{\mathsf{rec}},\ \mathbf{y}_{\mathsf{rec}} \in [-B, B]^m,\ \mathbf{z}_{\mathsf{rec}} \in [-\beta m B, \beta m B]^{\bar{m}},\ \mathbf{w} \in \{0,1\}^m, \\ \mathbf{s}_{0,\mathsf{oa}} \in \{0,1\}^{nk},\ \mathbf{x}_{\mathsf{oa}},\ \mathbf{y}_{\mathsf{oa}} \in [-B, B]^m,\ \mathbf{z}_{\mathsf{oa}} \in [-\beta m B, \beta m B]^{\bar{m}} \end{cases}$$

such that the following system of 10 equations holds:

$$\begin{cases} \mathbf{u} = [\mathbf{A}|\mathbf{A}_0|\mathbf{A}_1|\ldots|\mathbf{A}_\ell] \cdot \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \tau[1] \cdot \mathbf{d}_2 \\ \vdots \\ \tau[\ell] \cdot \mathbf{d}_2 \end{pmatrix} + (-\mathbf{D}) \cdot \mathbf{w}_{\mathsf{U}} \bmod q, \\ \mathbf{0} = \mathbf{H}_{n,q-1} \cdot \mathbf{w}_{\mathsf{U}} + (-\mathbf{D}_0) \cdot \mathbf{r} + (-\mathbf{D}_1) \cdot \mathbf{t}_{\mathsf{U}} \bmod q, \\ \mathbf{0} = \mathbf{H}_{2n,q-1} \cdot \mathbf{t}_{\mathsf{U}} + (-\mathbf{F}) \cdot \mathbf{b}_{\mathsf{U}} \bmod q, \\ \mathbf{c}^{(1)}_{\mathsf{rec}} = (\bar{\mathbf{A}}^\top \cdot \mathbf{H}_{n,q-1}) \cdot \mathbf{s}_{0,\mathsf{rec}} + \mathbf{I}_m \cdot \mathbf{y}_{\mathsf{rec}} \bmod q, \\ \mathbf{c}^{(2)}_{\mathsf{rec}} = \mathbf{Q} \cdot \mathbf{z}_{\mathbf{\Psi}} + (\mathbf{G}^\top \cdot \mathbf{H}^\top_{\mathsf{VK}} \cdot \mathbf{H}_{n,q-1}) \cdot \mathbf{s}_{0,\mathsf{rec}} + \mathbf{I}_{\bar{m}} \cdot \mathbf{z}_{\mathsf{rec}} \bmod q, \\ \mathbf{c}^{(3)}_{\mathsf{rec}} = (\mathbf{U}^\top \cdot \mathbf{H}_{n,q-1}) \cdot \mathbf{s}_{0,\mathsf{rec}} + \mathbf{I}_m \cdot \mathbf{x}_{\mathsf{rec}} + (\lfloor \frac{q}{2} \rfloor \cdot \mathbf{I}_m) \cdot \mathbf{w} \bmod q, \\ \mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q, \\ \mathbf{c}^{(1)}_{\mathsf{oa}} = (\bar{\mathbf{A}}^\top \cdot \mathbf{H}_{n,q-1}) \cdot \mathbf{s}_{0,\mathsf{oa}} + \mathbf{I}_m \cdot \mathbf{y}_{\mathsf{oa}} \bmod q, \\ \mathbf{c}^{(2)}_{\mathsf{oa}} = [(\mathbf{B}_{\mathsf{OA}} + \mathbf{H}_{\mathsf{VK}} \cdot \mathbf{G})^\top \cdot \mathbf{H}_{n,q-1}] \cdot \mathbf{s}_{0,\mathsf{oa}} + \mathbf{I}_{\bar{m}} \cdot \mathbf{z}_{\mathsf{oa}} \bmod q, \\ \mathbf{c}^{(3)}_{\mathsf{oa}} = (\mathbf{V}^\top \cdot \mathbf{H}_{n,q-1}) \cdot \mathbf{s}_{0,\mathsf{oa}} + \mathbf{I}_m \cdot \mathbf{x}_{\mathsf{oa}} + (\lfloor \frac{q}{2} \rfloor \cdot \mathbf{I}_m) \cdot \mathbf{t}_{\mathsf{U}} \bmod q. \end{cases} \tag{18}$$

Let $\mathbf{w}_1 = \mathbf{b}_{\mathsf{U}}$, $\mathbf{w}_2 = \mathbf{s}_{0,\mathsf{rec}}$, $\mathbf{w}_3 = \mathbf{z}_{\mathbf{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}})$, $\mathbf{w}_4 = \mathbf{w}_{\mathsf{U}}$, $\mathbf{w}_5 = \mathbf{t}_{\mathsf{U}}$, $\mathbf{w}_6 = \mathbf{s}_{0,\mathsf{oa}}$, $\mathbf{w}_7 = \mathbf{w}$, $\mathbf{w}_8 = \mathbf{x}_{\mathsf{rec}}$, $\mathbf{w}_9 = \mathbf{y}_{\mathsf{rec}}$, $\mathbf{w}_{10} = \mathbf{z}_{\mathsf{rec}}$, $\mathbf{w}_{11} = \mathbf{r}$, $\mathbf{w}_{12} = \mathbf{x}_{\mathsf{oa}}$, $\mathbf{w}_{13} = \mathbf{y}_{\mathsf{oa}}$, $\mathbf{w}_{14} = \mathbf{z}_{\mathsf{oa}}$ and

$$\mathbf{w}_{15} = \left(\mathbf{d}_1^\top \,\|\, \mathbf{d}_2^\top \,\|\, \tau[1] \cdot \mathbf{d}_2^\top \,\|\, \ldots \,\|\, \tau[\ell] \cdot \mathbf{d}_2^\top \right)^\top.$$

Then system (18) can be rewritten as:

$$\begin{cases} \mathbf{v}_1 = \mathbf{M}_{1,1} \cdot \mathbf{w}_1 + \mathbf{M}_{1,2} \cdot \mathbf{w}_2 + \ldots + \mathbf{M}_{1,15} \cdot \mathbf{w}_{15} \bmod q, \\ \mathbf{v}_2 = \mathbf{M}_{2,1} \cdot \mathbf{w}_1 + \mathbf{M}_{2,2} \cdot \mathbf{w}_2 + \ldots + \mathbf{M}_{2,15} \cdot \mathbf{w}_{15} \bmod q, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mathbf{v}_{10} = \mathbf{M}_{10,1} \cdot \mathbf{w}_1 + \mathbf{M}_{10,2} \cdot \mathbf{w}_2 + \ldots + \mathbf{M}_{10,15} \cdot \mathbf{w}_{15} \bmod q, \end{cases} \tag{19}$$

where $\{\mathbf{M}_{i,j}\}_{(i,j)\in[10]\times[15]}$, $\{\mathbf{v}_i\}_{i\in[10]}$ are public matrices and vectors (which are possibly zero).

The argument system is obtained by invoking the protocol from Section 4.2. The protocol is repeated κ times to make the soundness error negligibly small.

5.2 Efficiency and Correctness

Efficiency. It can be seen that the given group encryption scheme can be implemented in polynomial time. We now will evaluate the bit-sizes of keys and ciphertext, as well as the communication cost of the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$.

– The public key of GM, as in (12), has bit-size $\mathcal{O}(\ell n^2 \log^2 q) = \widetilde{\mathcal{O}}(\ell\lambda^2)$.
– The public keys of OA and each user both have bit-size $n\bar{m}\lceil \log_2 q \rceil = \widetilde{\mathcal{O}}(\lambda^2)$.
– The secret key of each party in the scheme is a trapdoor of bit-size $\widetilde{\mathcal{O}}(\lambda^2)$. The user's certificate $\mathsf{cert}_{\mathsf{U}}$ has bit-size $\widetilde{\mathcal{O}}(\lambda)$.
– The ciphertext $\boldsymbol{\Psi}$ consists of $\mathsf{VK} \in \mathbb{Z}_q^n$, two ABB ciphertexts of total size $2(2m+\bar{m})\lceil \log_2 q \rceil$ and a one-time signature Σ. Thus, its bit-size is $\widetilde{\mathcal{O}}(\lambda)+\big|\Sigma\big|$.
– The communication cost of the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ is largely dominated by the bit-size of the witness $\mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}}) \in \{0,1\}^{4n\bar{m}k^2}$. The total cost is $\kappa \cdot \mathcal{O}(n^2 \log^4 q) = \widetilde{\mathcal{O}}(\lambda^2)$ bits.

Correctness. The given group encryption scheme is correct with overwhelming probability. We first remark that the scheme parameters are set up so that the two instances of the ABB identity-based encryption [1] are correct. Indeed, during the decryption procedure of $\mathsf{DEC}(\mathsf{sk}_{\mathsf{U}}, \boldsymbol{\Psi}, L)$, we have:

$$\mathbf{c}_{\mathsf{rec}}^{(3)} - \mathbf{E}_{\mathsf{VK}}^{\top} \cdot \begin{bmatrix} \mathbf{c}_{\mathsf{rec}}^{(1)} \\ \mathbf{c}_{\mathsf{rec}}^{(2)} \end{bmatrix} = \mathbf{x}_{\mathsf{rec}} - \mathbf{E}_{\mathsf{VK}}^{\top} \cdot \begin{bmatrix} \mathbf{y}_{\mathsf{rec}} \\ \mathbf{z}_{\mathsf{rec}} \end{bmatrix} + \mathbf{w} \cdot \left\lfloor \frac{q}{2} \right\rfloor .$$

Note that $\|\mathbf{x}_{\mathsf{rec}}\|_\infty$ and $\|\mathbf{y}_{\mathsf{rec}}\|_\infty$ are bounded by B, and $\|\mathbf{z}_{\mathsf{rec}}\|_\infty = \|\mathbf{R}_{\mathsf{rec}}^{\top} \cdot \mathbf{y}_{\mathsf{rec}}\|_\infty \leq \beta m B = \widetilde{\mathcal{O}}(n^2)$. Furthermore, the entries of the discrete Gaussian matrix $\mathbf{E}_{\mathsf{VK}}^{\top}$ are bounded by $\widetilde{\mathcal{O}}(\sqrt{n})$. Hence, the error term $\mathbf{x}_{\mathsf{rec}} - \mathbf{E}_{\mathsf{VK}}^{\top} \cdot \begin{bmatrix} \mathbf{y}_{\mathsf{rec}} \\ \mathbf{z}_{\mathsf{rec}} \end{bmatrix}$ is bounded by $\widetilde{\mathcal{O}}(n^{3.5})$ which is much smaller than $q/4 = \widetilde{\mathcal{O}}(n^4)$. As a result, the decryption algorithm returns $\mathbf{w}$ with overwhelming probability. The correctness of algorithm $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \boldsymbol{\Psi}, L)$ also follows from a similar argument.

Finally, we note that if a certified group user honestly follows all the prescribed algorithms, then he should be able to compute valid witness-vectors to be used in the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$, and he should be accepted by the verifier, thanks to the perfect completeness of the argument system in Section 4.2.

5.3 Security

Our scheme is proven secure under the SIS and LWE assumptions using classical reduction techniques. The security results are explicited in the following theorems, for which some proofs have been deferred to Appendix C.

Theorem 2. *The scheme provides anonymity if the* LWE *assumption holds and if* $\mathcal{OTS}$ *is a strongly unforgeable one-time signature.* (The proof is given in Appendix C.1.)

Theorem 3. *The scheme provides message secrecy assuming that the* LWE *assumption holds and that* $\mathcal{OTS}$ *is a strongly unforgeable one-time signature.* (The proof is presented in Appendix C.2.)

Theorem 4. *The scheme provides soundness under the* SIS *assumption.*

Proof. To prove the result, we observe that, in order to break the soundness property, the adversary must come up with a relation $\mathsf{pk}_{\mathcal{R}} = (\mathbf{A}_R, \mathbf{u}_R) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$, a ciphertext $\Psi^\star = (\mathsf{VK}^\star, \mathbf{c}_{\mathsf{rec}}^\star, \mathbf{c}_{\mathsf{oa}}^\star, \Sigma^\star)$, a label L and produce a convincing proof $\pi_{\Psi^\star}$ such that either

a. $\mathbf{c}_{\mathsf{oa}}^\star$ does not decrypt to a string $\mathbf{h} \in \{0,1\}^m$ such that $\mathbf{h}_{\mathsf{U}} = \mathbf{H}_{2n,q-1} \cdot \mathbf{h} \in \mathbb{Z}_q^{2n}$ coincides with $\mathbf{h}_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,m,q}(\mathbf{B}_{\mathsf{U}}^\top)$ for some $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ appearing in $\mathsf{database}$.

b. $\mathbf{c}_{\mathsf{oa}}^\star$ opens to a certified public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$, which belongs to $\mathsf{database}$ (and for which a certificate was issued), but $\mathbf{B}_{\mathsf{U}}$ is outside the language $\mathcal{PK}$ of valid public keys. This case is immediately ruled out by the density of the public key space. Namely, all matrices $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ are potentially valid public keys as there always exist a small-norm matrix $\mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m \times \bar{m}}$ such that $\mathbf{B}_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} \bmod q$.

c. $\mathbf{c}_{\mathsf{oa}}^\star$ opens to a certified key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$ for which $\Psi^\star = (\mathsf{VK}^\star, \mathbf{c}_{\mathsf{rec}}^\star, \mathbf{c}_{\mathsf{oa}}^\star, \Sigma^\star)$ is not a valid encryption of $\mathbf{w} \in \{0,1\}^m$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$.

d. The opening algorithm fails to uniquely identify the receiver. This occurs if $\mathbf{c}_{\mathsf{oa}}^\star$ decrypts to a string $\mathbf{h} \in \{0,1\}^m$ such that $\mathbf{h}'_{\mathsf{U}} = \mathbf{H}_{2n,q-1} \cdot \mathbf{h} \in \mathbb{Z}_q^{2n}$ corresponds to at least two distinct public keys $\mathbf{B}_{\mathsf{U},0}, \mathbf{B}_{\mathsf{U},1} \in \mathbb{Z}_q^{n \times \bar{m}}$ which satisfy

$$\mathbf{h}'_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},0}^\top) \bmod q = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},1}^\top) \bmod q.$$

Since $\mathsf{mdec}_{n,\bar{m},q}(.) : \mathbb{Z}_q^{\bar{m} \times n} \to \{0,1\}^{n\bar{m}k}$ is an injective function, the above equality necessarily implies a collision for the SIS-based hash function built upon $\mathbf{F} \in \mathbb{Z}_q^{2n \times n\bar{m}k}$: namely,

$$\mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},0}^\top) - \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},1}^\top) \quad \in \{-1,0,1\}^{n\bar{m}k}$$

is a short non-zero vector of $\Lambda_q^\perp(\mathbf{F})$.

Having shown that cases b and d cannot occur if the SIS assumption holds, we only need to consider cases a and c. The computational soundness of the argument system ensures that, by replaying the soundness adversary a sufficient number of times, the knowledge extractor will be able to extract either: (i) A breach in the computational soundness of the argument system and thus the binding property of the commitment scheme COM (which relies on the SIS assumption with the commitment scheme of [32]). Note that this situation covers case (c.) above. (ii) A set of witnesses

$$\begin{cases} \tau \in \{0,1\}^\ell,\ \mathbf{d} = [\mathbf{d}_1^\top | \mathbf{d}_2^\top]^\top \in [-\beta,\beta]^{2m},\ \mathbf{r} \in [-\beta,\beta]^m \\ \mathbf{t}_{\mathsf{U}} \in \{0,1\}^m,\ \mathbf{w}_{\mathsf{U}} \in \{0,1\}^{\bar{m}} \\ \mathbf{b}_{\mathsf{U}} \in \{0,1\}^{n\bar{m}k},\ \mathbf{s}_{0,\mathsf{rec}} \in \{0,1\}^{nk},\ \mathbf{z}_{\mathbf{\Psi}} \in \{0,1\}^{4n\bar{m}k^2} \\ \mathbf{x}_{\mathsf{rec}},\ \mathbf{y}_{\mathsf{rec}} \in [-B,B]^m,\ \mathbf{z}_{\mathsf{rec}} \in [-\beta mB, \beta mB]^{\bar{m}},\ \mathbf{w} \in \{0,1\}^m, \\ \mathbf{s}_{\mathsf{oa}} \in \{0,1\}^{nk},\ \mathbf{x}_{\mathsf{oa}},\ \mathbf{y}_{\mathsf{oa}} \in [-B,B]^m,\ \mathbf{z}_{\mathsf{oa}} \in [-\beta mB, \beta mB]^{\bar{m}} \end{cases}$$

satisfying relations (18). Given that witnesses $\tau \in \{0,1\}^\ell$, $\mathbf{d} \in [-\beta,\beta]^{2m}$, $\mathbf{r} \in [-\beta,\beta]^m$ and $\mathbf{t}_{\mathsf{U}} \in \{0,1\}^m$ satisfy (18), it comes that $(\tau, \mathbf{d}, \mathbf{r})$ form a valid

signature for the message $\mathbf{t}_{\mathsf{U}} \in \{0,1\}^m$. At this point, case a implies that no matrix $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ of $\mathsf{database}$ decomposes to a string $\mathbf{h}_{\mathsf{U}} \in \{0,1\}^{n\bar{m}k}$ such that $\mathbf{t}_{\mathsf{U}} = \mathsf{vdec}_{n,q-1}(\mathbf{F} \cdot \mathbf{h}_{\mathsf{U}} \bmod q)$ was signed by the reduction during an execution of JOIN. This implies that the pair $\big(\mathbf{t}_{\mathsf{U}}, (\tau, \mathbf{d}, \mathbf{r})\big)$ forms a forgery for the SIS-based signature scheme of Appendix A.1. The reduction is straightforward and omitted. □

Acknowledgements

We thank Damien Stehlé for useful discussions and the reviewers for useful comments. 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". Huaxiong Wang was also supported by NTU under Tier 1 grant RG143/14.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010.
2. C. Aguilar-Melchor, S. Bettaieb, X. Boyen, L. Fousse, and P. Gaborit. Adapting lyubashevsky's signature schemes to the ring signature setting. In *AFRICACRYPT 2013*, volume 7918 of *LNCS*, pages 1–25. Springer, 2013.
3. L. E. Aimani and M. Joye. Toward practical group encryption. In *ACNS 2013*, volume 7954 of *LNCS*, pages 237–252. Springer, 2013.
4. M. Ajtai. Generating hard instances of the short basis problem. In *ICALP 1999*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
5. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS 2009*, volume 3 of *LIPIcs*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.
6. W. Banaszczyk. New bounds in some transference theorems in the geometry of number. *Mathematische Annalen*, 296(1):625–635, 1993.
7. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 566–582. Springer, 2001.
8. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS 1993*, pages 62–73. ACM Press, 1993.
9. 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 in LNCS, pages 551–572. Springer, 2014.
10. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS 2015*, volume 9326 of *LNCS*, pages 305–325. Springer, 2015.

11. 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.
12. 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.
13. 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.
14. 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.
15. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, number 2576 in LNCS, pages 268–289. Springer, 2002.
16. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
17. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.
18. J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 179–196. Springer, 2009.
19. D. Chaum and E. Van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
20. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 418–430. Springer, 2000.
21. 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*, volume 9452 of *LNCS*, pages 260–285. Springer, 2015.
22. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.
23. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 2009*, pages 169–178. ACM, 2009.
24. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206. ACM, 2008.
25. O. Goldreich, S. Goldwasser, and S. Halevi. Collision-Free Hashing from Lattice Problems. *ECCC*, 3(42), 1996.
26. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 1985*, pages 291–304. ACM, 1985.
27. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In *CRYPTO 2015*, number 9216 in LNCS, pages 503–523. Springer, 2015.
28. 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.
29. 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.
30. M. Izabachène, D. Pointcheval, and D. Vergnaud. Mediated traceable anonymous encryption. In *LATINCRYPT 2010*, volume 6212 of *LNCS*, pages 40–60. Springer, 2010.

31. 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.
32. 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.
33. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 571–589. Springer, 2004.
34. A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. In *ASIACRYPT 2007*, number 4833 in LNCS, pages 181–199. Springer, 2007.
35. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *EUROCRYPT 2005*, number 3494 in LNCS, pages 198–214. Springer, 2005.
36. 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.
37. 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.
38. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*. Springer, 2016.
39. 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*, volume 9666 of *LNCS*, pages 1–31. Springer, 2016.
40. B. Libert, M. Yung, M. Joye, and T. Peters. Traceable group encryption. In *PKC 2014*, volume 8383 of *LNCS*, pages 592–610. Springer, 2014.
41. 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.
42. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC 2015*, volume 9020 of *LNCS*, pages 427–449. Springer, 2015.
43. V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
44. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
45. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 282–298. Springer, 2003.
46. 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.
47. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT 1999*, number 1592 in LNCS, pages 223–238. Springer, 1999.
48. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC 2009*, pages 333–342. ACM, 2009.
49. C. Peikert and V. Vaikuntanathan. Non-interactive statistical zero-knowledge proofs for lattice problems. In *CRYPTO 2008*, volume 5157 of *LNCS*, pages 536–553. Springer, 2008.

50. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.
51. M. Rückert. Lattice-based blind signatures. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 413–430. Springer, 2010.
52. C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 239–252. Springer, 1989.
53. J. Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6):1757–1768, 1996.
54. M. Trolin and D. Wikström. Hierarchical group signatures. In *ICALP 2005*, volume 3580 of *LNCS*, pages 446–458. Springer, 2005.
55. 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 Building Blocks

A.1 Signatures Supporting Zero-Knowledge Proofs

We use a signature scheme proposed by Libert, Ling, Mouhartem, Nguyen and Wang [38] who extended the Böhl *et al.* signature [11] (which is itself built upon Boyen's signature [13]) into a signature scheme compatible with zero-knowledge proofs. While the scheme was designed to sign messages comprised of multiple blocks, we only use the single-block version here.

Keygen$(1^\lambda, q, n, m, \ell, \sigma)$**:** This algorithm takes as input a security parameter $\lambda > 0$ as well as the following parameters: $n = \mathcal{O}(\lambda)$; a prime modulus $q = \widetilde{\mathcal{O}}(n^4)$; dimension $m = 2n\lceil \log q \rceil$; an integer $\ell = \mathsf{poly}(\lambda)$; and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q} \log n)$. It defines the message space as $\{0,1\}^m$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_A}$ 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 $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m/2})$, $\mathbf{D}_0, \mathbf{D}_1 \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ as well as a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

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

$$PK := \big(\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{D}_0,\ \mathbf{D}_1,\ \mathbf{D},\ \mathbf{u}\big).$$

Sign$\big(SK, \mathfrak{m}\big)$**:** To sign a message $\mathfrak{m} \in \{0,1\}^m$,

1. Choose a random binary string $\tau \hookleftarrow U(\{0,1\}^\ell)$. Then, using $SK := \mathbf{T_A}$, compute a short delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the matrix

$$\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}. \tag{20}$$

2. Choose a discrete Gaussian vector $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^m, \sigma}$. Compute $\mathbf{c}_M \in \mathbb{Z}_q^n$ as a chameleon hash of $\mathfrak{m}$. Namely, compute $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathfrak{m} \in \mathbb{Z}_q^n$, which is used to define $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{c}_M) \in \mathbb{Z}_q^n$. 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 $sig = (\tau, \mathbf{v}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$.

Verify$\big(PK, \mathfrak{m}, sig\big)$**:** Given PK, $\mathfrak{m} \in \{0,1\}^m$ and $sig = (\tau, \mathbf{v}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$, return 1 if $\|\mathbf{v}\| < \sigma\sqrt{2m}$, $\|\mathbf{r}\| < \sigma\sqrt{m}$ and

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathfrak{m}) \bmod q. \tag{21}$$

Like [13,11], the scheme of [38] was proved secure under the SIS assumption and shown to easily interact with Stern-like protocols when it comes to proving knowledge of a hidden message-signature pair. While such proofs would also be possible using Boyen's signature [13], the number of public matrices $\{\mathbf{A}_j\}_{j=0}^\ell$ in the public key can be reduced from $\Theta(n \cdot \log q)$ to $\Theta(\lambda)$ using the scheme of [38].

The above description uses a slightly different variant of [38] where, at step 2 of the signing algorithm, a different binary decomposition of $\mathbf{c}_M$ is used to compute $\mathbf{u}_M$: while [38] uses the standard binary decomposition, we use a non-unique encoding based on the vdec function for convenience. However, the security proof of [38] goes through with this encoding since the function $\mathsf{vdec}_{n,q-1}(.)$ is injective.

Lemma 6 ([38, Th. 1]). *The above signature scheme is unforgeable under chosen-message attacks if the* SIS *assumption holds.*

A.2 The Agrawal-Boneh-Boyen IBE Scheme

Identity-Based Encryption. An IBE scheme is a tuple of efficient algorithms $(\mathsf{Setup}, \mathsf{Extract}_{\mathsf{PP}}, \mathsf{Encrypt}_{\mathsf{PP}}, \mathsf{Decrypt}_{\mathsf{PP}})$ such that

Setup(1^λ)**:** On security parameter λ, this algorithm outputs public parameters PP and a master secret key msk.

Extract$_{\mathsf{PP}}(\mathsf{msk}, \mathsf{ID})$**:** Takes as input a master secret key msk and an identity ID and outputs a secret key $\mathsf{sk}_{\mathsf{ID}}$.

Encrypt$_{\mathsf{PP}}(\mathsf{ID}, M)$**:** Given an identity ID and a message M, it outputs a ciphertext C.

Decrypt$_{\mathsf{PP}}(\mathsf{sk}_{\mathsf{ID}}, C)$**:** Given a secret key $\mathsf{sk}_{\mathsf{ID}}$ and a ciphertext C, outputs either a decryption error symbol $\perp$, or a message M.

Correctness requires that, for any pair $(\mathsf{PP}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, any ID and any message M, we have $\mathsf{Decrypt}_{\mathsf{PP}}\big(\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk}, \mathsf{ID}), \mathsf{Encrypt}_{\mathsf{PP}}(\mathsf{ID}, M)\big) = M$.

Our proofs rely on the semantic security of the scheme against selective adversaries (IND-sID-CPA) but also on the stronger property of ciphertext pseudorandomness. Informally, this notions demands that the adversary be unable to distinguish an encryption of a message of its choice from a random element of the ciphertext space $\mathcal{C}$. Notice that this property implies IND-sID-CPA security.

Definition 7. *An IBE scheme has pseudo-random-ciphertexts if no PPT adversary $\mathcal{A}$ with access to private key extraction oracle $\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk}, \cdot)$ has non-negligible advantage* $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{ROR}}(\lambda) = |\Pr\big[\mathbf{Expt}_{\mathcal{A}}^{\mathrm{ROR}} = 1\big] - \frac{1}{2}|$ *in this game:*

Experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathrm{ROR}}(\lambda)$

$\mathsf{ID}^\star \leftarrow \mathcal{A}_{id}(\lambda)$; $(\mathsf{PP}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$;
$M \leftarrow \mathcal{A}_{Ch}^{\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk},\cdot)}(\mathsf{PP})$;
$b \hookleftarrow U(\{0,1\})$;
if $b = 1$ *then* $C^\star \leftarrow \mathsf{Encrypt}_{\mathsf{PP}}(M, \mathsf{ID}^\star)$ *else* $C^\star \leftarrow U(\mathcal{C})$;
$b' \leftarrow \mathcal{A}_{guess}^{\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk},\cdot)}(C^\star)$;
if $b = b'$ *then return* 1 *else return* 0;

The ABB System. Agrawal, Boneh and Boyen described [1] a compact IBE scheme in the standard model which allows encrypting multi-bit messages.

$\mathsf{Setup}(1^\lambda)$: Given a security parameter λ, choose parameters q, n, σ, α and define $k = \lfloor \log q \rfloor$, $\bar{m} = nk$, $m = 2\bar{m}$ and choose a noise distribution χ for LWE.

1. Compute $(\bar{\mathbf{A}}, \mathbf{T}_{\bar{\mathbf{A}}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. Define $\mathbf{G} = \mathbf{I}_n \otimes [1|2|\ldots|2^{k-1}] \in \mathbb{Z}_q^{n\times\bar{m}}$. Sample matrices $\mathbf{B} \hookleftarrow U(\mathbb{Z}_q^{n\times\bar{m}})$, $\mathbf{U} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$.
3. Let $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ be the full-rank difference mapping from [1].

Output $\mathsf{PP} = \left(\bar{\mathbf{A}}, \mathbf{B}, \mathbf{U}\right)$ and $\mathsf{msk} = \mathbf{T}_{\bar{\mathbf{A}}}$.

$\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk}, \mathsf{ID})$: Given $\mathsf{msk} = \mathbf{T}_{\bar{\mathbf{A}}}$ and an identity $\mathsf{ID} \in \mathbb{Z}_q^n$, do as follows:

1. Define the matrix $\mathbf{B}_{\mathsf{ID}} = \mathbf{B} + \mathsf{FRD}(\mathsf{ID}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n\times\bar{m}}$.
2. Let $\mathbf{B}_{\mathbf{A},\mathsf{ID}} = [\mathbf{A} \mid \mathbf{B}_{\mathsf{ID}}] \in \mathbb{Z}_q^{n\times(m+\bar{m})}$, use $\mathbf{T}_A$ to compute a delegated basis $\mathbf{T}_{\mathsf{ID}}$ for the lattice $\Lambda^\perp(\mathbf{B}_{\mathbf{A},\mathsf{ID}})$.
3. Use $\mathbf{T}_{\mathsf{ID}}$ to sample a small-norm matrix $\mathbf{E}_{\mathsf{ID}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ satisfying the equality $\mathbf{B}_{\mathbf{A},\mathsf{ID}} \cdot \mathbf{E}_{\mathsf{ID}} = \mathbf{U} \bmod q$.
4. Output $\mathsf{sk}_{\mathsf{ID}} = \mathbf{E}_{\mathsf{ID}} \in \mathbb{Z}^{(m+\bar{m})\times m}$.

$\mathsf{Encrypt}_{\mathsf{PP}}(\mathsf{ID}, \mathbf{m})$: Given an identity ID and a message $\mathbf{m} \in \{0,1\}^m$,

1. Compute the matrix $\mathbf{B}_{\mathsf{ID}} = \mathbf{B} + \mathsf{FRD}(\mathsf{ID}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n\times\bar{m}}$. Sample vectors $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n), \mathbf{x}, \mathbf{y} \hookleftarrow \chi^m$, $\mathbf{R} \hookleftarrow D_{\mathbb{Z},\sigma}^{m\times\bar{m}}$ and compute $\mathbf{z} = \mathbf{R}^\top \cdot \mathbf{y} \in \mathbf{Z}^m$.
2. Compute
$$\begin{cases} \mathbf{c}^{(1)} = \bar{\mathbf{A}}^\top \cdot \mathbf{s} + \mathbf{y} \bmod q, \\ \mathbf{c}^{(2)} = \mathbf{B}_{\mathsf{ID}}^\top \cdot \mathbf{s} + \mathbf{z} \bmod q, \\ \mathbf{c}^{(3)} = \mathbf{U}^\top \cdot \mathbf{s} + \mathbf{x} + \mathbf{m} \cdot \left\lfloor \frac{q}{2} \right\rfloor. \end{cases} \tag{22}$$
3. Output $\mathbf{c} = \left(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mathbf{c}^{(3)}\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$.

$\mathsf{Decrypt}_{\mathsf{PP}}(\mathsf{sk}_{\mathsf{ID}}, \mathbf{c})$: Given $\mathsf{sk}_{\mathsf{ID}} = \mathbf{E}_{\mathsf{ID}}$ and $\mathbf{c} = \left(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mathbf{c}^{(3)}\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$, compute and output $\mathbf{m}' = \left\lfloor \left(\mathbf{c}^{(3)} - \mathbf{E}_{\mathsf{ID}} \cdot \begin{bmatrix} \mathbf{c}^{(1)} \\ \mathbf{c}^{(2)} \end{bmatrix} \right) \cdot \left\lfloor \frac{q}{2} \right\rfloor^{-1} \right\rceil \in \{0,1\}^m$.

Theorem 5 ([1, Th. 23]). *The ABB IBE scheme has pseudo-random ciphertexts if the $\mathsf{LWE}_{n,q,\chi}$ assumption holds.*

B Proof of Theorem 1

We first restate Theorem 1.

Theorem 6. *The protocol in Figure 1 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(D \log q)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{M}, \mathbf{v})$*, 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* $\mathbf{M} \cdot \mathbf{x}' = \mathbf{v} \bmod q$*.*

Proof. It can be checked that the protocol has perfect completeness: 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 $\widetilde{\mathcal{O}}(D \log q)$.

We now prove that the protocol is a statistical zero-knowledge argument of knowledge.

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 $\overline{Ch} \in \{1, 2, 3\}$ as 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 $\mathbf{x}' \in \mathbb{Z}_q^D$ such that $\mathbf{M} \cdot \mathbf{x}' = \mathbf{v} \bmod q$. Next, it samples $\mathbf{r}_x \leftarrow U(\mathbb{Z}_q^D)$, $\pi \leftarrow U(\mathcal{S})$, 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

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

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

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

Case $\overline{Ch} = 2$: SIM samples $\mathbf{x}' \leftarrow U(\mathsf{VALID})$, $\mathbf{r}_x \leftarrow U(\mathbb{Z}_q^D)$, $\pi \leftarrow U(\mathcal{S})$, 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

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

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

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

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

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

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 $\mathrm{RSP}_1 = (\mathbf{t}_x, \mathbf{t}_r, \rho_2^{(1)}, \rho_3^{(1)})$, $\mathrm{RSP}_2 = (\pi_2, \mathbf{y}_2, \rho_1^{(2)}, \rho_3^{(2)})$, $\mathrm{RSP}_3 = (\pi_3, \mathbf{y}_3, \rho_1^{(3)}, \rho_2^{(3)})$ are 3 valid responses to the same commitment CMT $= (C_1, C_2, C_3)$, with respect to all 3 possible values of the challenge. The validity of these responses implies that:

$$\begin{cases} \mathbf{t}_x \in \mathsf{VALID}; \\ C_1 = \mathsf{COM}(\pi_2, \mathbf{M} \cdot \mathbf{y}_2 - \mathbf{v}; \rho_1^{(2)}) = \mathsf{COM}(\pi_3, \mathbf{M} \cdot \mathbf{y}_3; \rho_1^{(3)}); \\ C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2^{(1)}) = \mathsf{COM}(\Gamma_{\pi_3}(\mathbf{y}_3); \rho_2^{(3)}); \\ C_3 = \mathsf{COM}(\mathbf{t}_x + \mathbf{t}_r; \rho_3^{(1)}) = \mathsf{COM}(\Gamma_{\pi_2}(\mathbf{y}_2); \rho_3^{(2)}). \end{cases}$$

Since COM is computationally binding, we can deduce that:

$$\mathbf{t}_x \in \mathsf{VALID}; \pi_2 = \pi_3; \mathbf{t}_r = \Gamma_{\pi_3}(\mathbf{y}_3); \mathbf{t}_x + \mathbf{t}_r = \Gamma_{\pi_2}(\mathbf{y}_2); \mathbf{M} \cdot \mathbf{y}_2 - \mathbf{v} = \mathbf{M} \cdot \mathbf{y}_3 \bmod q.$$

Let $\mathbf{x}' = \mathbf{y}_2 - \mathbf{y}_3$, then we have $\Gamma_{\pi_2}(\mathbf{x}') = \mathbf{t}_x \in \mathsf{VALID}$ which implies that $\mathbf{x}' \in \mathsf{VALID}$. Furthermore, we have $\mathbf{M} \cdot \mathbf{x}' = \mathbf{M} \cdot (\mathbf{y}_2 - \mathbf{y}_3) = \mathbf{v} \bmod q$.

This concludes the proof. □

C Security Proofs for the Group Encryption Scheme

C.1 Anonymity (Proof of Theorem 2)

Proof. We consider a sequence of games where the first game is the real experiment of definition 5 while, in the final game, the adversary $\mathcal{A}$ is essentially an adversary against the anonymity of the Agrawal-Boneh-Boyen IBE scheme [1]. In Game i, we call W_i the event that the challenger outputs 1.

Game 1**:** The challenger $\mathcal{B}$ generates public parameters param, which include matrices $\bar{\mathbf{A}}, \mathbf{U}, \mathbf{V} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{F} \in \mathbb{Z}_q^{2n\times n\bar{m}k}$. The opening authority's public key $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$ is given to $\mathcal{A}$ who generates a group manager's public key $\mathsf{pk}_{\mathsf{GM}}$ of its own. By invoking the USER oracle, $\mathcal{A}$ registers two distinct receivers' public keys $\mathsf{pk}_{\mathsf{U},0} = \mathbf{B}_{\mathsf{U},0} \in \mathbb{Z}_q^{n\times\bar{m}}$, $\mathsf{pk}_{\mathsf{U},1} = \mathbf{B}_{\mathsf{U},1} \in \mathbb{Z}_q^{n\times\bar{m}}$ chosen by the challenger. It also makes a number of opening queries and decryption queries, which the challenger handles using $\mathsf{sk}_{\mathsf{OA}} = \mathbf{T}_{\mathsf{OA}}$ and $\mathsf{sk}_{\mathsf{U},0} = \mathbf{T}_{\mathsf{U},0}$, $\mathsf{sk}_{\mathsf{U},1} = \mathbf{T}_{\mathsf{U},1}$, respectively. After a while, the adversary outputs $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$, with $\mathbf{A}_R \in \mathbb{Z}_p^{n\times m}$, $\mathbf{u}_R \in \mathbb{Z}_q^n$ and $\mathbf{w} \in \{0,1\}^m$. In return, $\mathcal{A}$ obtains, as a challenge, a group encryption $\Psi^\star = (\mathsf{VK}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{oa}}, \Sigma^\star)$. of the witness $\mathbf{w}$ under $\mathsf{pk}_{\mathsf{U},b} = \mathbf{B}_{\mathsf{U},b}$, for some random bit $b \leftarrow U(\{0,1\})$ of the challenger's choice. Then, the adversary obtains proofs $\pi^\star_{\Psi^\star}$ for $\Psi^\star$ and makes further opening and decryption queries under the natural restrictions of Definition 5. When the adversary $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$ and the challenger outputs 1 if and only if $b' = b$.

Game 2**:** This game is like Game 1 except the challenger aborts in the event that the adversary $\mathcal{A}$ queries the opening of a ciphertext $\Psi = (\mathsf{VK}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, \Sigma)$ such that $\mathsf{VK} = \mathsf{VK}^\star$ and σ is valid (we assume w.l.o.g. that $\mathsf{VK}^\star$ is generated ahead of time). If this event occurs, the adversary $\mathcal{A}$ is necessarily able to break the strong unforgeability of $\mathcal{OTS}$ (note that, if the query occurs before the challenge phase, it means that $\mathcal{A}$ has forged a signature without seeing a signature at all). There thus exist a one-time signature forger $\mathcal{B}$ such that $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathrm{ots}}_{\mathcal{B}}(\lambda)$, which means that Game 2 is identical to Game 1 so long as $\mathcal{OTS}$ is a strongly unforgeable one-time signature.

Game 3**:** In this game, we modify the generation of proofs $\pi^\star_{\Psi^\star}$: instead of generating proofs using the real witnesses, we appeal to the zero-knowledge simulator of the argument system of Section 4.2 at each invocation of $\mathcal{P}$ after the challenge phase. Note that, since we assume public parameters generated by a trusted party, the statistical ZK simulator is allowed to use a trapdoor embedded in param to generate simulated proofs (using, e.g., Damgård's technique [20]). The statistical zero-knowledge property of the argument system ensures that $\mathcal{A}$'s view remains statistically close to that of Game 2: we have $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$.

Game 4**:** We now modify the generation of the challenge ciphertext $\Psi^\star$. In this game, the challenger computes the ciphertext $\mathbf{c}^\star_{\mathsf{oa}}$ as an ABB encryption under the identity $\mathsf{VK}^\star$ of a random m-bit string instead of a decomposition

$\mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U},b}) \in \{0,1\}^m$ of $\mathbf{h}_{\mathsf{U},b} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},b}^{\top}) \in \mathbb{Z}_q^{2n}$. Since the random encryption coins $\mathbf{s}_{\mathsf{oa}}^{\star}, \mathbf{R}_{\mathsf{oa}}^{\star}, \mathbf{x}_{\mathsf{oa}}^{\star}, \mathbf{y}_{\mathsf{oa}}^{\star}$ are no longer used to generate proofs $\pi_{\Psi^\star}$, we can show that any noticeable change in $\mathcal{A}$'s output distribution implies a selective adversary against the ABB IBE, as established by Lemma 7, which would contradict the LWE assumption. The result of Agrawal *et al.* [1, Theorem 23] (recalled in Theorem 5) indeed implies $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

In Game 4, we can show that, if the adversary $\mathcal{A}$ has noticeable advantage in the anonymity game, we can break the anonymity of the ABB IBE system, as shown in the proof of Lemma 8. From the result of [1, Theorem 23], we deduce that $|\Pr[W_4] - 1/2| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$, which implies the announced result. □

Lemma 7. *Any PPT adversary such that* $\Pr[W_4]$ *is noticeably different from* $\Pr[W_3]$ *implies a selective adversary against the ABB IBE scheme.*

Proof. Let $\mathcal{A}$ be a PPT adversary for which $|\Pr[W_4] - \Pr[W_3]| = \varepsilon$ is non-negligible. We use $\mathcal{A}$ to build a selective adversary against the ABB IBE system.

At the outset of the game, the reduction $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{VK}^\star, \mathsf{SK}^\star)$ and declares $\mathsf{VK}^\star$ as the target identity to its challenger for the selective security game, and obtains in return the IBE public parameters

$$\mathsf{PP} = \big(\bar{\mathbf{A}}, \mathbf{B}, \mathbf{V}\big) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times \bar{m}} \times \mathbb{Z}_q^{n\times m}.$$

Next, the reduction runs the appropriate steps of the actual $\mathsf{SETUP}_{\mathsf{init}}$ algorithm to obtain $\mathsf{COM}_{\mathsf{par}}$, $\mathbf{F} \in \mathbb{Z}_q^{2n\times n\bar{m}k}$ and $\mathbf{U} \in \mathbb{Z}_q^{n\times m}$. Namely, $\mathcal{B}$ samples $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{2m\times n\bar{m}k})$ and $\mathbf{U} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ like in the $\mathsf{SETUP}_{\mathsf{init}}$ algorithm and sends

$$\mathsf{param} = \big\{\lambda, n, q, k, m, B, \chi, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\big\}$$

along with $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B} \in \mathbb{Z}_q^{n\times \bar{m}}$ to the adversary $\mathcal{A}$.

In return, the adversary $\mathcal{A}$ chooses $\mathsf{pk}_{\mathsf{GM}}$, which allows it to enroll two users for whom $\mathcal{B}$ faithfully generates $(\mathsf{pk}_{\mathsf{U},i}, \mathsf{sk}_{\mathsf{U},i})_{i\in\{0,1\}}$. Knowing both private keys $\{\mathsf{sk}_{\mathsf{U},i} = \mathbf{T}_{\mathsf{U},i}\}_{i\in\{0,1\}}$, $\mathcal{B}$ is able to perfectly simulate the $\mathsf{DEC}(\cdot)$ oracle.

Open Queries. To answer opening queries for ciphertexts $\mathbf{\Psi} = (\mathsf{VK}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, \Sigma)$ and labels L, $\mathcal{B}$ first checks that $\mathsf{Ver}(\mathsf{VK}, \Sigma, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, L)) = 1$. If this test fails, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ queries its IBE challenger to obtain a IBE private key $\mathbf{T}_{\mathsf{OA},\mathsf{VK}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ for identity $\mathsf{VK} \neq \mathsf{VK}^\star$. The IBE challenger's response allows $\mathcal{B}$ to decrypt $\mathbf{c}_{\mathsf{OA}}$ and figure out the identity of the receiver by looking up $\mathsf{database}$. The result of the opening operation is then returned to $\mathcal{A}$.

After a number of queries, $\mathcal{A}$ decides to move to the challenge phase and sends a challenge query $\big((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}^\star, L^\star\big)$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w}^\star \bmod q$. The reduction handles this query by requesting a challenge ciphertext for the IBE security game with the messages $\mathbf{m}_0 = \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U},b})$, for some random bit $b \hookleftarrow U(\{0,1\})$ and $\mathbf{m}_1 \leftarrow U(\{0,1\}^m)$. In return, $\mathcal{B}$ obtains a challenge ciphertext $\mathbf{c}_{\mathsf{OA}}^\star$ under identity $\mathsf{VK}^\star$, which is embedded in $\mathcal{A}$'s challenge ciphertext.

Namely, $\mathbf{\Psi}^\star = (\mathsf{VK}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{OA}}, \Sigma^\star)$ is obtained by computing $\mathbf{c}^\star_{\mathsf{rec}}$ as an ABB encryption of the witness $\mathbf{w}^\star$ using the matrix $\mathbf{B}_{\mathsf{U},b} \in \mathbb{Z}_q^{n\times\bar{m}}$ as in (15) and $\Sigma^\star = \mathsf{Sign}(\mathsf{SK}^\star, (\mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{OA}}, L^\star))$. All queries to the proving oracle $\mathcal{P}$ are replied by returning a simulated ZK argument as in Game 3.

When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$. If $b = b'$, $\mathcal{B}$ returns the bit 0 as a guess that the selective security challenger encrypted $\mathbf{m}_0 = \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U},b})$. Otherwise, $\mathcal{B}$ outputs 1 meaning that the IBE challenger chose to encrypt $\mathbf{m}_1$, which was chosen independently of the value of $b \in \{0,1\}$. If we call Random (resp. Real) the event that the IBE challenger chooses to encrypt $\mathbf{m}_1$ (resp. $\mathbf{m}_0$), we can assess the advantage of the reduction $\mathcal{B}$ as

$$\begin{aligned}\mathbf{Adv}^{\text{sID-CPA}}_{\mathcal{B}}(\lambda) &= |\Pr[b = b' \mid \mathsf{Random}] - \Pr[b = b' \mid \mathsf{Real}]| \\ &= |\Pr[W_4] - \Pr[W_3]| \\ &= \varepsilon,\end{aligned}$$

which proves the result. □

Lemma 8. *In Game* 4*, the adversary's advantage is negligible assuming that the ABB IBE has pseudo-random ciphertexts.*

Proof. Let us assume the existence of a PPT adversary $\mathcal{A}$ with non negligible advantage ε in Game 4. From $\mathcal{A}$, we construct a selective adversary $\mathcal{B}$ that can distinguish ABB ciphertexts from random elements of the ciphertext space with non-negligible advantage in the game described in Definition 7.

First, $\mathcal{B}$ generates $(\mathsf{SK}^\star, \mathsf{VK}^\star)$ via the key generation algorithm of the one-time-signature OTS and hands $\mathsf{VK}^\star$ to its pseudo-randomness challenger. In return, $\mathcal{B}$ receives

$$\mathsf{PP} = \big(\bar{\mathbf{A}}, \mathbf{B}, \mathbf{U}\big) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times\bar{m}} \times \mathbb{Z}_q^{n\times m}$$

from its real-or-random (ROR) challenger.

Our reduction uses PP to compute public parameters for our GE scheme. To this end, it samples $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{2n\times n\bar{m}k})$, $\mathbf{V} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ as in the real $\mathsf{SETUP}_{\mathsf{init}}$ algorithm. The reduction $\mathcal{B}$ also computes $\mathbf{B}_{\mathsf{OA}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{OA}} \bmod q$, where the small-norm matrix $\mathbf{T}_{\mathsf{OA}}$ is sampled from $D^{m\times\bar{m}}_{\mathbb{Z},\sigma}$, and sends $\mathcal{A}$ the parameters

$$\mathsf{param} = \big\{\lambda, n, q, k, m, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\big\},$$

where $\bar{\mathbf{A}}$ is taken from PP, along with $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}}$. The rest of the keys are generated as in Game 4.

The reduction $\mathcal{B}$ then tosses a coin $b \hookleftarrow U(\{0,1\})$. When the adversary $\mathcal{A}$ triggers an execution of the join protocol, $\mathcal{B}$ generates the public keys $(\mathsf{pk}_i)_{i\in\{0,1\}}$ by defining $\mathsf{pk}_{\mathsf{U},b} = \mathbf{B}$ using the matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times\bar{m}}$ supplied by the ROR challenger as part of PP and generates $(\mathsf{pk}_{\mathsf{U},1-b}, \mathsf{sk}_{1-b}) = (\mathbf{B}_{\mathsf{U},1-b} = \bar{\mathbf{A}} \cdot \mathbf{T}_{1-b}, \mathbf{T}_{1-b})$ for a secret key $\mathbf{T}_{1-b} \leftarrow D^{\bar{m}}_{\mathbb{Z}^m,\sigma}$ of its own. The two public keys $(\mathsf{pk}_{\mathsf{U},i})_{i\in\{0,1\}}$ are then certified by the adversarially-controlled GM. Notice that in the adversary's view, both public keys $\mathsf{pk}_{\mathsf{U},b}$ and $\mathsf{pk}_{\mathsf{U},1-b}$ are identically distributed.

To answer decryption queries $(\mathbf{\Psi} = (\mathsf{VK}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, \Sigma), L)$, for any query pertaining to $\mathsf{pk}_{\mathsf{U},b}$, the reduction invokes its ROR challenger to obtain an IBE private key for the identity $\mathsf{VK} \neq \mathsf{VK}^\star$ and uses the result to decrypt $\mathbf{c}_{\mathsf{rec}}$. For any decryption query involving $\mathsf{pk}_{\mathsf{U},1-b}$, the reduction can faithfully run the actual decryption algorithm using its trapdoor $\mathbf{T}_{1-b}$. Open queries are answered using $\mathbf{T}_{\mathsf{OA}}$ as in the real Open algorithm.

When the adversary $\mathcal{A}$ decides to do so, it queries a challenge for a triple $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ of its choice subject to the constraint $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w}$. At this point, $\mathcal{B}$ queries a challenge to its own challenger for the message $\mathbf{w}$ and obtains a ciphertext $\mathbf{c}$, which is embedded in $\mathbf{\Psi}^\star = (\mathsf{VK}^\star, \mathbf{c}, \mathbf{c}^\star_{\mathsf{OA}}, \Sigma^\star)$ while $\mathbf{c}^\star_{\mathsf{OA}}$ and $\Sigma^\star$ are generated as in Game 3 (in particular, $\mathbf{c}^\star_{\mathsf{OA}}$ encrypts a random string instead of a hash value of $\mathsf{pk}_{\mathsf{U},b}$). After the challenge phase, all queries to the proving oracle $\mathcal{P}$ are replied by returning a simulated ZK argument as in Game 3.

When $\mathcal{A}$ ends, it outputs a bit $b' \in \{0,1\}$. If $b' = b$, the reduction outputs Real. Otherwise, it outputs Random. Indeed, if the ROR challenger is playing the real game, we are exactly in Game 4: we have $\Pr[b' = b|\mathsf{Real}] = \Pr[W_4]$. Otherwise, the challenge ciphertext $\mathbf{\Psi}^\star$ is completely independent of $b \in \{0,1\}$ so that we can only have $b' = b$ with probability $\Pr[b' = b|\mathsf{Random}] = 1/2$. It follows that $\mathbf{Adv}^{\mathrm{ROR}}_{\mathcal{B}}(\lambda) \geq |\Pr[W_4] - 1/2|$. □

C.2 Message Secrecy (Proof of Theorem 3)

Proof. We proceed via a sequence of games. The first one corresponds to the experiment of Definition 4 when the challenger's bit b is 1 and the adversary obtains an actual encryption of the witness $\mathbf{w} \in \{0,1\}^m$ and real proofs at each invocation of the $\mathsf{PROVE}(.)$ oracle. In the last game, the adversary $\mathcal{A}$ is given an encryption of some random plaintext whereas $\mathsf{PROVE}(.)$ returns simulated zero-knowledge arguments which are generated a simulator $\mathcal{P}'$ that does not use any witness. In Game i, W_i stands for the event that the adversary $\mathcal{A}$ outputs the bit $b' = 1$.

Game 1: This is the real game, where the challenger feeds $\mathcal{A}$ with public parameters param containing $\bar{\mathbf{A}}, \mathbf{U}, \mathbf{V} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{F} \in \mathbb{Z}_q^{2n \times n\bar{m}k}$. The adversary produces public keys $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$ and $\mathsf{pk}_{\mathsf{GM}} = (\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{D}, \mathbf{u})$ on behalf of the opening authority and the group manager which are both under its control. The challenger and $\mathcal{A}$ run an execution of the JOIN protocol which allows $\mathcal{A}$ to register and certify the public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ of some honest receiver chosen by the challenger. Then, the adversary $\mathcal{A}$ makes a polynomial number of decryption queries which the challenger faithfully handles using the private key $\mathsf{sk}_{\mathsf{U}} = \mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m \times \bar{m}}$ for which $\mathbf{B}_{\mathsf{U}} \cdot \mathbf{T}_{\mathsf{U}} = \mathbf{0}^{n \times \bar{m}}$. At some point, the adversary $\mathcal{A}$ outputs a triple $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$, with $\mathbf{A}_R \in \mathbb{Z}_p^{n \times m}$, $\mathbf{u}_R \in \mathbb{Z}_q^n$ and $\mathbf{w} \in \{0,1\}^m$. At this point, the challenger generates a challenge ciphertext $\Psi^\star = (\mathsf{VK}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{oa}}, \Sigma^\star)$ consisting of a group encryption of the real witness $\mathbf{w}$ under $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$. Then, the adversary obtains a polynomial number of proofs $\pi^\star_{\Psi^\star}$ related to the challenge ciphertext $\Psi^\star$ and

is granted further access to the decryption oracle under the obvious restrictions. When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$.

Game 2: In this game, we modify the $\mathsf{DEC}(.)$ oracle and have the challenger reject any ciphertext of the form $\Psi = (\mathsf{VK}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, \Sigma)$ such that $\mathsf{VK} = \mathsf{VK}^\star$ (note that $\mathsf{VK}^\star$ can be generated at the outset of the game w.l.o.g.). Clearly Game 2 is identical to Game 1 until the event that the challenger rejects a ciphertext that would not have been rejected in Game 1. This can only occur if $\mathcal{A}$ is able to break the strong unforgeability of the one-time signature $\mathcal{OTS}$. As in the proof of Theorem 2, we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$, which is negligible if $\mathcal{OTS}$ is strongly unforgeable.

Game 3: We now modify the generation of proofs $\pi^\star_{\Psi^\star}$. Instead of generating them using the witnesses used in the generation of $\Psi^\star$, we rely on the zero-knowledge simulator of the argument system of Section 4.2 at each invocation of $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}$ after the challenge phase (note that, since we assume trusted public parameters, the simulator can use techniques [20] to achieve statistically perfect simulation without increasing the number of rounds). The statistical ZK property of the argument system ensures that this change will remain unnoticed, even in the view of an all powerful adversary: we have $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$. From now onwards, the random coins $coins^\star_{\Psi} = \left(\mathbf{s}^\star_{\mathsf{rec}}, \mathbf{R}^\star_{\mathsf{rec}}, \mathbf{x}^\star_{\mathsf{rec}}, \mathbf{y}^\star_{\mathsf{rec}}, \mathbf{s}^\star_{\mathsf{oa}}, \mathbf{R}^\star_{\mathsf{oa}}, \mathbf{x}^\star_{\mathsf{oa}}, \mathbf{y}^\star_{\mathsf{oa}}\right)$ are no longer used by the PROVE oracle.

Game 4: In the generation of $\Psi^\star$, we set $\mathbf{c}^\star_{\mathsf{rec}}$ as an encryption of a random element of $\mathbb{Z}_p^m$. Since the random encryption coins $\mathbf{s}^\star_{\mathsf{rec}}, \mathbf{R}^\star_{\mathsf{rec}}, \mathbf{x}^\star_{\mathsf{rec}}, \mathbf{y}^\star_{\mathsf{rec}}$ are not used in Game 3, Lemma 9 gives a simple reduction showing that any significant change in $\mathcal{A}$'s behavior would imply a selective adversary against the ABB identity-based encryption scheme. The result of [1] tells us that, under the LWE assumption, Game 4 is computationally indistinguishable from Game 3 in the adversary's view: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

Game 5: We bring a last modification to the $\mathsf{DEC}(.)$ oracle and now refrain from applying the rejection rule of Game 2. If $\mathcal{OTS}$ is strongly unforgeable, the distance $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$ must be negligible.

In the last game, the oracle $\mathsf{PROVE}(.)$ does not need to know any witness. It thus mirrors the experiment of Definition 4 where the challenger's bit is $b = 0$. Putting everything altogether, we get $|\Pr[W_5] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$, which yields the claimed result. □

Lemma 9. *Any PPT adversary that can distinguish Game* 4 *from Game* 3 *implies a selective adversary against the ABB IBE scheme.*

Proof. Let us assume a PPT adversary $\mathcal{A}$ such that $\varepsilon = \left|\Pr[W_4] - \Pr[W_3]\right|$ is noticeable. We use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that breaks the IND-sID-CPA security of the ABB scheme, which would contradict the LWE assumption, as established in [1, Th. 23].

At the very beginning of the IND-sID-CPA game, the reduction $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{SK}^\star, \mathsf{VK}^\star)$ and hands $\mathsf{VK}^\star$ to its selective security

challenger as the target identity under which the challenge ciphertext will later be computed. In response, $\mathcal{B}$ receives the public parameters

$$\mathsf{PP} = (\bar{\mathbf{A}}, \mathbf{B}, \mathbf{U}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times \bar{m}} \times \mathbb{Z}_q^{n\times m}$$

from its IBE challenger.

The reduction then runs the missing steps of the actual $\mathsf{Setup}_{\mathsf{init}}$ algorithm: namely, $\mathcal{B}$ samples $\mathbf{F} \leftarrow U(\mathbb{Z}_q^{2m\times n\bar{m}k}), \mathbf{V} \leftarrow U(\mathbb{Z}_q^{n\times m})$ and generates $\mathsf{COM}_{\mathsf{par}}$ before sending the common public parameters

$$\mathsf{param} = \big\{\lambda, n, q, k, m, B, \chi, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\big\}$$

to the adversary $\mathcal{A}$.

At this point, the adversary $\mathcal{A}$ chooses the public keys $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times \bar{m}}$ and $\mathsf{pk}_{\mathsf{GM}} = (\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{D}, \mathbf{u})$ on behalf of the opening authority and the group manager. It also starts an execution of the joining protocol in which the reduction $\mathcal{B}$ defines $\mathsf{pk}_{\mathsf{U}} = \mathbf{B} \in \mathbb{Z}_q^{n\times \bar{m}}$ as the honest receiver's public key, where $\mathbf{B} \in \mathbb{Z}_q^{n\times \bar{m}}$ is taken from the public parameters PP supplied by its IBE challenger. Note that $\mathsf{pk} = \mathbf{B} \in \mathbb{Z}_q^{n\times \bar{m}}$ is distributed as a real key in $\mathcal{A}$'s view. This public key is certified by $\mathbf{A}$ which controls the GM.

In the next stage, $\mathcal{A}$ makes a number of decryption queries for ciphertexts of the form $\mathbf{\Psi} = (\mathsf{VK}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, \Sigma)$. To answer these, the reduction invokes its IBE challenger so as to obtain an IBE private key $\mathbf{E}_{\mathsf{VK}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ for the identity $\mathsf{VK} \neq \mathsf{VK}^{\star}$. The resulting $\mathbf{E}_{\mathsf{VK}}$ is used it to IBE-decrypt $\mathbf{c}_{\mathsf{rec}}$ and return the corresponding witness $\mathbf{w}$ to $\mathbf{A}$.

At some point, the adversary $\mathcal{A}$ queries a challenge ciphertext by outputting a triple $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ such that $\mathbf{w} \in \{0,1\}^m$ satisfies $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$. Then, the reduction $\mathcal{B}$ requests a challenge ciphertext $\mathbf{c}^{\star}_{\mathsf{rec}}$ to its IBE challenger by sending it the messages $\mathbf{m}_1 = \mathbf{w} \in \{0,1\}^m$ and $\mathbf{m}_0 \leftarrow U(\{0,1\}^m)$. The resulting ciphertext $\mathbf{c}^{\star}_{\mathsf{rec}}$ is embedded in $\mathbf{\Psi}^{\star} = (\mathsf{VK}^{\star}, \mathbf{c}^{\star}_{\mathsf{rec}}, \mathbf{c}^{\star}_{\mathsf{OA}}, \Sigma^{\star})$ by faithfully computing $\mathbf{c}^{\star}_{\mathsf{OA}}$ and $\Sigma^{\star}$ as in the actual Enc algorithm.

After the challenge phase, $\mathcal{A}$ keeps sending decryption queries for ciphertexts $\Psi^{\star}$ containing one-time verification keys $\mathsf{VK} \neq \mathsf{VK}^{\star}$ and these decryption queries are answered as before. In addition, $\mathcal{A}$ is granted access to the stateful oracle $\mathsf{PROVE}^{b}_{\mathcal{P},\mathcal{P}'}$. Recall that, from Game 3 onwards, all these queries are answered by returning simulated zero-knowledge arguments. Eventually $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is also returned by $\mathcal{B}$ to its own challenger.

If the IBE challenger provides a challenge $\mathbf{c}^{\star}_{\mathsf{rec}}$ that encrypts a random message (i.e., by encrypting $\mathbf{m}_0$), then we are exactly in the setting of Game 4. In the even that $\mathbf{c}^{\star}_{\mathsf{rec}}$ rather encrypts $\mathbf{m}_1 = \mathbf{w} \in \{0,1\}^m$, $\mathcal{A}$'s view is exactly the same as in Game 3. If we denote by Random (resp. Real) the event that the IBE challenger chooses to encrypt $\mathbf{m}_0$ (resp. $\mathbf{m}_1$), the advantage of the reduction $\mathcal{B}$ as an IND-sID-CPA adversary is

$$\begin{aligned}\mathbf{Adv}^{\text{sID-CPA}}_{\mathcal{B}}(\lambda) &= |\Pr[b' = 1|\mathsf{Real}] - \Pr[b' = 1|\mathsf{Random}]| = |\Pr[W_3] - \Pr[W_4]| \\ &= \varepsilon,\end{aligned}$$

which concludes our proof. □