# Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures without Trapdoors

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

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

**Abstract.** An accumulator is a function that hashes a set of inputs into a short, constant-size string while preserving the ability to efficiently prove the inclusion of a specific input element in the hashed set. It has proved useful in the design of numerous privacy-enhancing protocols, in order to handle revocation or simply prove set membership. In the lattice setting, currently known instantiations of the primitive are based on Merkle trees, which do not interact well with zero-knowledge proofs. In order to efficiently prove the membership of some element in a zero-knowledge manner, the prover has to demonstrate knowledge of a hash chain without revealing it, which is not known to be efficiently possible under well-studied hardness assumptions. In this paper, we provide an efficient method of proving such statements using involved extensions of Stern's protocol. Under the Small Integer Solution assumption, we provide zero-knowledge arguments showing possession of a hash chain. As an application, we describe new lattice-based group and ring signatures in the random oracle model. In particular, we obtain: (i) The first lattice-based ring signatures with logarithmic size in the cardinality of the ring; (ii) The first lattice-based group signature that does not require any GPV trapdoor and thus allows for a much more efficient choice of parameters.

## 1 Introduction

Cryptographic accumulators were introduced by Benaloh and de Mare [10] as alternative to digital signatures in the design of distributed protocols. While initially used in time-stamping and membership testing mechanisms [10], they found numerous applications in the context of fail-stop signatures [7], anonymous credentials [20,19,1,44], group signatures [68], anonymous *ad hoc* authentication [28], digital cash [6,22,54], set membership proofs [69,63] or authenticated data structures [60,59] (see [27] for further examples).

In a nutshell, an accumulator is a sort of algebraic hash function that maps a large set $R$ of inputs into a short, constant-size accumulator value $u$ such that an efficiently computable short witness $w$ provides evidence that a given input was indeed incorporated into the hashed set. In order to be useful, the size of the witness should be much smaller than the cardinality of the input set. An extension, suggested by Camenisch and Lysyanskaya [20], allows the accumulator value to be updated over time, by adding or deleting elements of the hashed set while preserving the ability to efficiently update witnesses. For most applications, the usual security requirement mandates the infeasibility of computing an accumulator value $u$ and a valid witness $w$ for an element $x$ outside the set of hashed inputs. This is made possible by public-key techniques like the existence of a trapdoor (e.g., the factorization of an RSA modulus or the discrete logarithm of some public group element) hidden behind public parameters.

So far, number theoretic realizations have been divided into two main families. The first one relies on groups of hidden order [10,7,47,15] and includes proposals based on the Strong RSA assumption [7,43]. The second main family [57,19] was first explored by Nguyen [57] and appeals to bilinear maps (a.k.a. pairings) and assumptions of variabe size like the Strong Diffie-Hellman assumption [14]. Strong-RSA-based candidates enjoy the advantage of short public parameters and they easily extend into universal accumulators [43] (where non-membership witnesses can show that a given input was not accumulated). While pairing-based schemes [57,19] usually require linear-size public parameters in the number of elements to be hashed, they are useful in applications [6,22] where we want to limit the number of elements to be hashed. A third family (e.g., [59]) of constructions relies on Merkle trees [50] rather than number theoretic assumptions. Its main disadvantage is that the use of hash trees makes it hardly compatible with efficient zero-knowledge proofs, which are inevitable ingredients of privacy-preserving protocols [20,68,19,1]. In fact, currently known methods [15,9] for reconciling Merkle trees and zero-knowledge proofs require non-standard assumptions in groups of hidden order [15] or the machinery of SNARKs, which inherently rely on non-falsifiable [55] knowledge assumptions [35].

Despite its wide range of applications, the accumulator primitive still has a relatively small number of efficient realizations. For the time being, most known solutions require non-standard *ad hoc* assumptions like Strong RSA or Strong Diffie-Hellman. To our knowledge, the only exception is a generic construction from vector commitments [24], which leaves open the problem of candidates based on the standard Computational Diffie-Hellman assumption (in groups without a bilinear map) or zero-knowledge-friendly lattice-based schemes. In this paper, we describe a new construction based on standard lattice assumptions which interacts nicely with zero-knowledge proofs despite the use of Merkle trees. We show that this new construction enables new, unexpected applications to the design of lattice-based ring signatures and group signatures.

Our Contributions. We describe a lattice-based accumulator[3] that enables short zero-knowledge arguments of membership. Our construction relies on a Merkle hash tree which is computed in a special way that makes it compatible

[3] A lattice-based accumulator was previously claimed in [38]. However, the generation of witnesses can only be performed using the secret key of the system. Moreover, their scheme is seemingly not compact due to the required choice of parameters.

with efficient protocols for proving possession of a secret value (i.e., a leaf of the tree) that is properly accumulated in the root of the tree. More specifically, our system allows demonstrating the knowledge of a hash chain from the considered secret leaf to the root in a zero-knowledge manner. This building block enables many interesting applications. In particular, we use it to design lattice-based ring and group signatures with dramatic improvements over the existing constructions. In the random oracle model, we obtain:

– The first lattice-based ring signature with logarithmic signature size in the cardinality of the ring. So far, all suggested proposals have linear size in the number of ring members.
– A lattice-based group signature with much shorter public key, signature length, and weaker hardness assumptions than all earlier realizations.

Our ring signature does not require any other setup assumption than having all users agree on a modulus $q$, a lattice dimension $n$ and a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ (which can be derived from a random oracle). It provably satisfies the strong security definitions put forth by Bender, Katz and Morselli [11].

Our group signature is analyzed in the setting of static groups using the definitions of Bellare, Micciancio and Warinschi [8]. Its salient feature (which it shares with our ring signature) is that, unlike all earlier candidates [33,41,42,46,58], it does not require the use of a trapdoor (as defined by Gentry, Peikert and Vaikuntanathan [31]) consisting of a short basis of some lattice. It thus eliminates one of the frequently cited reasons [49] for which lattice-based signatures tend to be impractical. In fact, our group signature departs from previously used design principles – which are all inspired in some way by the general construction of [8] – in that, surprisingly, it does not even require an ordinary digital signature to begin with. All we need is a lattice-based accumulator with a compatible zero-knowledge argument system for arguing knowledge of a hash chain.

Our Techniques. Our accumulator proceeds by computing a Merkle tree using a hash function based on the Small Integer Solution ($\mathsf{SIS}$) problem, which is a variant of the hash functions considered in [4,32,53] previously considered by Papamanthou *et al.* [59]. Instead of hashing a vector $\mathbf{x} \in \{0,1\}^m$ by computing its syndrome $\mathbf{A} \cdot \mathbf{x} \in \mathbb{Z}_q^n$ via a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, it outputs the coordinate-wise binary decomposition $\mathsf{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q) \in \{0,1\}^{m/2}$ of the syndrome to obtain the two-fold compression factor that is needed for iteratively applying the function in a Merkle tree. However, Papamanthou *et al.* [59] did not consider the problem of proving knowledge of a hash chain in a zero-knowledge fashion. The main technical novelty that we introduce is thus a method for demonstrating knowledge of a Merkle-tree hash chain using the framework of Stern's protocol [67].

Using this method, we build ring and group signatures with logarithmic size in the number of ring or group members involved. Our constructions are conceptually simple. Each user's private key is a random $m$-bit vector $\mathbf{x} \in \{0,1\}^m$ and the matching public key is the binary expansion $\mathbf{d} = \mathsf{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q) \in \{0,1\}^{m/2}$ of the corresponding syndrome. In order to sign a message, the user considers an accumulation $\mathbf{u} \in \{0,1\}^{m/2}$ of all users' public keys $R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1})$ – which is obtained by dynamically forming the ring $R$ in the ring signature and simply consists of the group public key in the group signature – and generates a Stern-type argument that: (i) His public key $\mathbf{d}_j$ belongs to the hashed set $R$; (ii) He knows the underlying secret $\mathbf{d}_j = \mathsf{bin}(\mathbf{A} \cdot \mathbf{x}_j \bmod q)$; (iii – for the group signature) He has honestly encrypted the binary representation of the integer $j$ determining his position in the tree to a ciphertext attached in the signature. In order to acquire anonymity in the strongest sense (i.e., where the adversary is granted access to a signature opening oracle), we apply the Naor-Yung paradigm [56] to Regev's cryptosystem [64], as was previously considered in [12]. As pointed out earlier, the advantage of not relying on an ordinary digital signature[4] lies in that it does not require any party (i.e., neither the group manager nor the group members in the case of group signatures) to have a GPV trapdoor [31] consisting of a short lattice basis. As emphasized by Lyubashevsky [49], explicitly avoiding the use of such trapdoors allows for drastically more efficient choices of parameters. As by-products, our scheme features much smaller group public key and users' secret keys, produces shorter signatures, and relies on weaker hardness assumptions than all of the existing lattice-based group signature schemes [33,21,41,46,58] in the BMW model [8].

In the following, we give an estimated efficiency comparison among our group signature and the previous 2 most efficient schemes with CCA-anonymity, by Ling *et al.* [46] and Nguyen *et al.* [58]. The estimations are done with parameter $n = 2^8$, group size $N = 1024$, and soundness error $2^{-80}$ for the $\mathsf{NIZKs}$.

– Ling *et al.*'s scheme requires $q = \mathcal{O}(\log N \cdot n^2)$, $m \geq 2n \log q$, so we set $q = 2^{18}$ and $m = 2^9 \cdot 18$. The infinity norm bound for discrete Gaussian samples is $2^6$. The scheme produces group public key size 65.8 MB; user's secret key size 13.5 KB (a Boyen signature [17]); and signature size 1.20 GB.
– Nguyen *et al.*'s scheme requires $q > m^{8.5}$, $m \geq 2n \log q$, so we set $q = 2^{142}$ and $m = 2^9 \cdot 142$. The scheme produces group public key size 2.15 GB; user's secret key size 90 GB (a trapdoor in $\mathbb{Z}^{3m \times 3m}$ with $(\log m)$-bit entries); and signature size 500 MB.
– Our scheme works with $q = 2^8$, $m = 2^9 \cdot 8$, and parameters $p = 32719$, $m_E = 7980$ for the encryption layer. The scheme features public key size 4.9 MB; user's secret key size 3.25 KB; and it produces signatures of size 61.5 MB.

Related Work. While originally suggested as a 3-move code-based identification scheme, Stern's protocol was adapted to the lattice setting by Kawachi *et al.* [40] and extended by Ling *et al.* [45] into an argument system for the Inhomogeneous Small Integer Solution ($\mathsf{ISIS}$) problem. In particular, Ling *et al.* gave a method, called *decomposition-extension* framework, which allows arguing knowledge of an integer vector $\mathbf{x} \in \mathbb{Z}^m$ of norm $\|\mathbf{x}\|_\infty \leq \beta$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \in \mathbb{Z}_q^n$ without leaving any gap between the vector computed by the knowledge extractor and the actual witness $\mathbf{x}$. As shown in [46], the technique of Ling *et al.* [45]

[4] Recall that all $\mathcal{O}(\log N)$-size group signatures employ a signature scheme in the standard model (for which all known constructions use trapdoors) in order to smoothly interact with zero-knowledge proofs.

can be used to prove more involved statements such as the possession of a Boyen signature [17] on a message encrypted by a dual Regev ciphertext [31]. Here, we take one step further and develop a zero-knowledge argument of knowledge (ZKAoK) that a specific element of some universe belongs to a hashed set.

Ring signatures were introduced by Rivest, Shamir and Tauman-Kalai [65] with the motivation of hiding the identity of a source (e.g., a whistleblower in a political scandal) while providing guarantees of trustworthiness. Bender, Katz and Morselli [11] gave stringent security definitions while constructions with sub-linear signature size were given by Chandran, Groth and Sahai [25]. The celebrated results of Gentry, Peikert and Vaikuntanathan [31] inspired a number of lattice-based ring signatures. The state-of-the-art construction probably stems from the framework of Brakerski and Tauman-Kalai [18], which results in linear-size in the number of ring members. The same holds for all known Fiat-Shamir-like lattice-based ring signatures (e.g., [40,2]), although some of them do not require a trapdoor. Thus far, the only logarithmic-size ring signatures [36,16] arise from the results of Groth and Kohlweiss [36] and it is not clear how to extend them to the lattice setting.

The notion of group signatures dates back to Chaum and Van Heyst [26]. While viable constructions were given in the seminal paper by Ateniese, Camenisch, Joye and Tsudik [5], their security notions remained poorly understood until the work of Bellare, Micciancio and Warinschi [8]. The first lattice-based proposal came out with the results of Gordon, Katz and Vaikuntanathan [33], which inspired a number of follow-up works describing new systems with a better asymptotic efficiency [41,58,46] or additional properties [21,42]. For the time being, the most efficient candidates are the recent concurrent proposals of Nguyen *et al.* and Ling *et al.* [58,46]. As it turns out, except for one scheme [12] that mixes lattice-based and discrete-logarithm-related assumptions, all currently available candidates [41,58,46,21,42] utilize a GPV trapdoor, either to perform the setup of the system or to trace signatures (or both). Our results thus provide the first system that completely eliminates GPV trapdoors.

At a high level, our ZKAoK system is partially inspired by the way Langlois *et al.* [42] made use of the Bonsai tree technique [23] since it proves knowledge of a solution to a SIS problem determined by the user's position in a tree. However, there are fundamental differences since our tree is built in a bottom-up (rather than top-down) manner and we do not perform any trapdoor delegation.

## 2 Preliminaries

Notations. We assume that all vectors are column vectors. The concatenation of matrices $\mathbf{A} \in \mathbb{Z}^{k \times i}$, $\mathbf{B} \in \mathbb{Z}^{k \times j}$ is denoted by $[\mathbf{A}|\mathbf{B}] \in \mathbb{Z}^{k \times (i+j)}$. For $b \in \{0,1\}$, we denote the bit $1-b \in \{0,1\}$ by $\bar{b}$. For a positive integer $i$, we let $[i]$ be the set $\{1,\ldots,i\}$. If $S$ is a finite set, $x \xleftarrow{\$} S$ means that $x$ is chosen uniformly at random from $S$. All logarithms are of base 2. The addition in $\mathbb{Z}_2$ is denoted by $\oplus$.

In this section, we first recall the average-case lattice problems SIS and LWE, together with their hardness results; and the notion of statistical zero-knowledge arguments of knowledge. The definitions and security requirements of cryptographic accumulators, ring signatures, and group signatures are deferred to their respective Sections 3, 4, and 5.

## 2.1 Average-case Lattice Problems

**Definition 1 ([3,31]).** The $\mathsf{SIS}^{\infty}_{n,m,q,\beta}$ problem is as follows: Given uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq \beta$ and $\mathbf{A}\cdot\mathbf{x} = \mathbf{0} \bmod q$.

If $m, \beta = \mathsf{poly}(n)$, and $q > \beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$, then the $\mathsf{SIS}^{\infty}_{n,m,q,\beta}$ problem is at least as hard as the worst-case lattice problem $\mathsf{SIVP}_\gamma$ for some $\gamma = \beta \cdot \widetilde{\mathcal{O}}(\sqrt{nm})$ (see [31,52]). Specifically, when $\beta = 1$, $q = \widetilde{\mathcal{O}}(n)$, $m = 2n\lceil \log q\rceil$, the $\mathsf{SIS}^{\infty}_{n,m,q,1}$ problem is at least as hard as $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$.

In the last decade, numerous $\mathsf{SIS}$-based cryptographic primitives have been proposed. In this work, we will extensively employ 2 such constructions:

– Our Merkle tree accumulator is built upon a specific family of collision-resistant hash functions, which is a syntactic modification (*i.e.*, it takes two inputs, instead of one) of the one presented in [3,53]. A similar scheme that works with larger $\mathsf{SIS}$ norm bound $\beta$ was proposed in [59].
– Our zero-knowledge argument systems use the statistically hiding and computationally binding string commitment scheme from [40].

For appropriate setting of parameters, the security of the above two constructions can be based on the worst-case hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$.

In the group signature in Section 5, we will employ the multi-bit version of Regev's encryption scheme [64], presented in [39][62]. The scheme is based on the hardness of the $\mathsf{LWE}$ problem.

**Definition 2 ([64]).** Let $n, m_E \geq 1$, $p \geq 2$, and let $\chi$ be a probability distribution on $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_p^n$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \hookleftarrow \chi$, and outputting $(\mathbf{a}, \mathbf{s}^\top \cdot \mathbf{a} + e) \in \mathbb{Z}_p^n \times \mathbb{Z}_p$. The $\mathsf{LWE}_{n,p,\chi}$ problem asks to distinguish $m_E$ samples chosen according to $\mathcal{A}_{\mathbf{s},\chi}$ (for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$) and $m_E$ samples chosen according to the uniform distribution over $\mathbb{Z}_p^n \times \mathbb{Z}_p$.

If $p$ is a prime power, $\chi$ is the discrete Gaussian distribution $D_{\mathbb{Z},\alpha p}$, where $\alpha p \geq 2\sqrt{n}$, then $\mathsf{LWE}_{n,p,\chi}$ is as least as hard as $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n/\alpha)}$ (see [64,61,51,52]).

## 2.2 Zero-Knowledge Arguments of Knowledge

We will work with statistical zero-knowledge argument systems, namely, interactive protocols where the zero-knowledge property holds against *any* cheating verifier, while the soundness property only holds against *computationally bounded* cheating provers. More formally, let the set of statements-witnesses $\mathrm{R} = \{(y, w)\} \in \{0,1\}^* \times \{0,1\}^*$ be an $\mathsf{NP}$ relation. A two-party game $\langle \mathcal{P}, \mathcal{V}\rangle$ is called an interactive argument system for the relation R with soundness error $e$ if the following two conditions hold:

– Completeness. If $(y, w) \in \mathrm{R}$ then $\Pr\big[\langle \mathcal{P}(y, w), \mathcal{V}(y)\rangle = 1\big] = 1$.
– Soundness. If $(y, w) \notin \mathrm{R}$, then $\forall$ PPT $\widehat{\mathcal{P}}$: $\Pr[\langle \widehat{\mathcal{P}}(y, w), \mathcal{V}(y)\rangle = 1] \le e$.

An argument system is called statistical zero-knowledge if for any $\widehat{\mathcal{V}}(y)$, there exists a PPT simulator $\mathcal{S}(y)$ producing a simulated transcript that is statistically close to the one of the real interaction between $\mathcal{P}(y, w)$ and $\widehat{\mathcal{V}}(y)$. A related notion is argument of knowledge, which requires the witness-extended emulation property. For protocols consisting of 3 moves (*i.e.*, commitment-challenge-response), witness-extended emulation is implied by *special soundness* [34], where the latter assumes that there exists a PPT extractor which takes as input a set of valid transcripts with respect to all possible values of the 'challenge' to the same 'commitment', and outputs $w'$ such that $(y, w') \in \mathrm{R}$.

The statistical zero-knowledge arguments of knowledge (sZKAoK) presented in this work are Stern-type [67]. In particular, they are $\Sigma$-protocols in the generalized sense defined in [37,12] (where 3 valid transcripts are needed for extraction, instead of just 2). Several recent works rely on Stern-type protocols to design lattice-based [45,42,46] and code-based [37,29] constructions.

## 3 A Lattice-Based Accumulator with Supporting Zero-Knowledge Argument of Knowledge

Throughout the paper, we will work with positive integers $n, q, k, m$, where: $n$ is the security parameter; $q = \widetilde{\mathcal{O}}(n)$; $k = \lceil \log q \rceil$; and $m = 2nk$. We identify $\mathbb{Z}_q$ by the set $\{0, \ldots, q-1\}$. We define the "powers-of-2" matrix

$$\mathbf{G} = \begin{bmatrix} 1\ 2\ 4\ \ldots\ 2^{k-1} & & & \\ & 1\ 2\ 4\ \ldots\ 2^{k-1} & & \\ & & \ldots & \\ & & & 1\ 2\ 4\ \ldots\ 2^{k-1} \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}.$$

Note that for every $\mathbf{v} \in \mathbb{Z}_q^n$, we have $\mathbf{v} = \mathbf{G} \cdot \mathsf{bin}(\mathbf{v})$, where $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{nk}$ denotes the binary representation of $\mathbf{v}$.

### 3.1 Cryptographic Accumulators

An *accumulator scheme* is a tuple of algorithms (TSetup, TAcc, TWitness, TVerify) defined as follows:

TSetup($n$) On input security parameter $n$, output the public parameter $pp$.

TAcc$_{pp}$ On input a set $R = \{\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}\}$ of $N$ data values, output an accumulator value $\mathbf{u}$.

TWitness$_{pp}$ On input a data set $R$ and a value $\mathbf{d}$, output $\perp$ if $\mathbf{d} \notin R$; otherwise output a witness $w$ for the fact that $\mathbf{d}$ is accumulated in TAcc($R$). (Typically, the size of $w$ should be short (*e.g.*, constant or logarithmic in $N$) to be useful.)

TVerify$_{pp}$ On input accumulator value $\mathbf{u}$ and a value-witness pair $(\mathbf{d}, w)$, output 1 (which indicates that $(\mathbf{d}, w)$ is valid for the accumulator $\mathbf{u}$) or 0.

An accumulator scheme is called correct if for all $pp \leftarrow \mathsf{TSetup}(n)$, we have $\mathsf{TVerify}_{pp}\big(\mathsf{TAcc}_{pp}(R), \mathbf{d}, \mathsf{TWitness}_{pp}(R, \mathbf{d})\big) = 1$ for all $\mathbf{d} \in R$.

The security of an accumulator scheme, as defined in [7,20], says that it is infeasible to prove that a value $\mathbf{d}^*$ was accumulated in a value $\mathbf{u}$ if it was not. This property is formalized as follows.

**Definition 3.** A*n accumulator scheme* $(\mathsf{TSetup}, \mathsf{TAcc}, \mathsf{TWitness}, \mathsf{TVerify})$ *is called secure if for all PPT adversaries* $\mathcal{A}$*:*

$$\Pr\big[pp \leftarrow \mathsf{TSetup}(n); (R, \mathbf{d}^*, w^*) \leftarrow \mathcal{A}(pp) : \\ \mathbf{d}^* \notin R \wedge \mathsf{TVerify}_{pp}(\mathsf{TAcc}_{pp}(R), \mathbf{d}^*, w^*) = 1\big] = \mathsf{negl}(n).$$

### 3.2 A Family of Lattice-Based Collision-Resistant Hash Functions

We now describe the specific family of lattice-based collision-resistant hash functions, upon which our Merkle hash tree will be built.

**Definition 4.** *The function family* $\mathcal{H}$ *mapping* $\{0,1\}^{nk} \times \{0,1\}^{nk}$ *to* $\{0,1\}^{nk}$ *is defined as* $\mathcal{H} = \{h_{\mathbf{A}} \mid \mathbf{A} \in \mathbb{Z}_q^{n\times m}\}$*, where for* $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1]$ *with* $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n\times nk}$*, and for any* $(\mathbf{u}_0, \mathbf{u}_1) \in \{0,1\}^{nk} \times \{0,1\}^{nk}$*, we have:*

$$h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathsf{bin}\big(\mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 \bmod q\big) \in \{0,1\}^{nk}.$$

Note that $h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{u} \Leftrightarrow \mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 = \mathbf{G} \cdot \mathbf{u} \bmod q$.

**Lemma 1.** *The function family* $\mathcal{H}$*, defined in 4 is collision-resistant, assuming the hardness of the* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *problem.*

*Proof.* Given $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1] \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, if one can find two *distinct* pairs $(\mathbf{u}_0, \mathbf{u}_1) \in \big(\{0,1\}^{nk}\big)^2$ and $(\mathbf{v}_0, \mathbf{v}_1) \in \big(\{0,1\}^{nk}\big)^2$ such that $h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = h_{\mathbf{A}}(\mathbf{v}_0, \mathbf{v}_1) \bmod q$, then one can obtain a *non-zero* vector $\mathbf{z} = \begin{pmatrix} \mathbf{u}_0 - \mathbf{v}_0 \\ \mathbf{u}_1 - \mathbf{v}_1 \end{pmatrix} \in \{-1,0,1\}^m$ such that

$$\mathbf{A} \cdot \mathbf{z} = \mathbf{A}_0 \cdot (\mathbf{u}_0 - \mathbf{v}_0) + \mathbf{A}_1 \cdot (\mathbf{u}_1 - \mathbf{v}_1) = \mathbf{G} \cdot h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) - \mathbf{G} \cdot h_{\mathbf{A}}(\mathbf{v}_0, \mathbf{v}_1) = \mathbf{0} \bmod q.$$

In other words, $\mathbf{z}$ is a valid solution to the $\mathsf{SIS}^{\infty}_{n,m,q,1}$ problem associated with matrix $\mathbf{A}$. The lemma then follows from the worst-case to average-case reduction from $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$. □

### 3.3 Our Merkle-Tree Accumulator

We now give the construction of a Merkle tree with $N = 2^\ell$ leaves, where $\ell$ is a positive integer, based on the family of lattice-based hash function $\mathcal{H}$ defined above.

$\mathsf{TSetup}(n)$. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and output $pp = \mathbf{A}$.

$\mathsf{TAcc}_{\mathbf{A}}(R = \{\mathbf{d}_0 \in \{0,1\}^{nk}, \ldots, \mathbf{d}_{N-1} \in \{0,1\}^{nk}\})$. For every $j \in [0, N-1]$, let $(j_1, \ldots, j_\ell) \in \{0,1\}^\ell$ be the binary representation of $j$, and let $\mathbf{d}_j = \mathbf{u}_{j_1,\ldots,j_\ell}$. Form the tree of depth $\ell = \log N$ based on the $N$ leaves $\mathbf{u}_{0,0,\ldots,0}, \ldots, \mathbf{u}_{1,1,\ldots,1}$ as follows:

1. At depth $i \in [\ell]$, the node $\mathbf{u}_{b_1,\ldots,b_i} \in \{0,1\}^{nk}$, for all $(b_1, \ldots, b_i) \in \{0,1\}^i$, is defined as $h_{\mathbf{A}}(\mathbf{u}_{b_1,\ldots,b_i,0}, \mathbf{u}_{b_1,\ldots,b_i,1})$.
2. At depth 0: The root $\mathbf{u} \in \{0,1\}^{nk}$ is defined as $h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1)$.

The algorithm outputs the accumulator value $\mathbf{u}$.

$\mathsf{TWitness}_{\mathbf{A}}(R, \mathbf{d})$. If $\mathbf{d} \notin R$, return $\perp$. Otherwise, $\mathbf{d} = \mathbf{d}_j$ for some $j \in [0, N-1]$ with binary representation $(j_1, \ldots, j_\ell)$. Output the witness $w$ defined as:

$$w = \big((j_1, \ldots, j_\ell), (\mathbf{u}_{j_1,\ldots,j_{\ell-1},\bar{j_\ell}}, \ldots, \mathbf{u}_{j_1,\bar{j_2}}, \mathbf{u}_{\bar{j_1}})\big) \in \{0,1\}^\ell \times \big(\{0,1\}^{nk}\big)^\ell,$$

for $\mathbf{u}_{j_1,\ldots,j_{\ell-1},\bar{j_\ell}}, \ldots, \mathbf{u}_{j_1,\bar{j_2}}, \mathbf{u}_{\bar{j_1}}$ computed by algorithm $\mathsf{TAcc}_{\mathbf{A}}(R)$.

$\mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u}, \mathbf{d}, w\big)$. Let the given witness $w$ be of the form:

$$w = \big((j_1, \ldots, j_\ell), (\mathbf{w}_\ell, \ldots, \mathbf{w}_1)\big) \in \{0,1\}^\ell \times \big(\{0,1\}^{nk}\big)^\ell.$$

The algorithm recursively computes the path $\mathbf{v}_\ell, \mathbf{v}_{\ell-1}, \ldots, \mathbf{v}_1, \mathbf{v}_0 \in \{0,1\}^{nk}$ as follows: $\mathbf{v}_\ell = \mathbf{d}$ and

$$\forall i \in \{\ell-1, \ldots, 1, 0\} : \; \mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases}$$

Then it returns 1 if $\mathbf{v}_0 = \mathbf{u}$. Otherwise, it returns 0.

In Figure 1, we give an illustrative example of a tree with $2^3 = 8$ leaves.

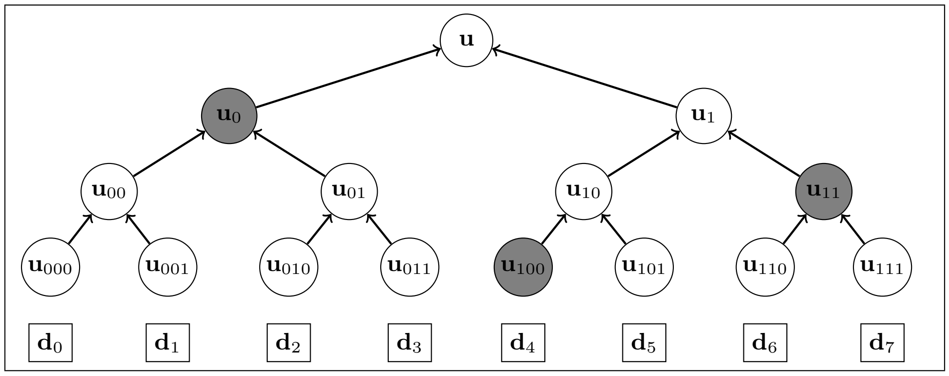

**Fig. 1:** A Merkle tree with $2^3 = 8$ leaves, which accumulates the data blocks $\mathbf{d}_0, \ldots, \mathbf{d}_7$ into the value $\mathbf{u}$ at the root. The bit string (101) and the gray nodes form a witness to the fact that $\mathbf{d}_5$ is accumulated in $\mathbf{u}$.

One can check that the above Merkle-tree accumulator scheme is correct. Furthermore, its security is based on the collision-resistance of the hash function family $\mathcal{H}$, which in turn is based on the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$.

**Theorem 1.** *The given accumulator scheme is secure in the sense of Definition 3, assuming the hardness of the* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *problem.*

*Proof.* Assuming that there exists a PPT adversary $\mathcal{B}$ who has non-negligible success probability in the security experiment of Definition 3. It receives a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ generated by $\mathsf{TSetup}(n)$, and returns $(R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}), \mathbf{d}^*, w^*)$ such that $\mathbf{d}^* \notin R$ and $\mathsf{TVerify}_{\mathbf{A}}(\mathbf{u}^*, \mathbf{d}^*, w^*) = 1$, where $\mathbf{u}^* = \mathsf{TAcc}_{\mathbf{A}}(R)$.

Parse $w^* = ((j_1^*, \ldots, j_\ell^*), (\mathbf{w}_\ell^*, \ldots, \mathbf{w}_1^*))$. Let $j^* \in [0, N{-}1]$ be the integer having binary representation $(j_1^*, \ldots, j_\ell^*)$ and let $\mathbf{u}_{j_1^*,\ldots,j_\ell^*} = \mathbf{d}_{j^*}, \mathbf{u}_{j_1^*,\ldots,j_{\ell-1}^*}, \ldots, \mathbf{u}_{j_1^*}, \mathbf{u}^*$ be the path from the leave $\mathbf{d}_{j^*}$ to the root of the tree generated by $\mathsf{TAcc}_{\mathbf{A}}(R)$. On the other hand, let $\mathbf{v}_\ell^* = \mathbf{d}^*, \mathbf{v}_{\ell-1}^*, \ldots, \mathbf{v}_1^*, \mathbf{v}_0^* = \mathbf{u}^*$ be the path computed by algorithm $\mathsf{TVerify}_{\mathbf{A}}(\mathbf{u}^*, \mathbf{d}^*, w^*)$. Note that $\mathbf{d}^* \neq \mathbf{d}_{j^*}$ since $\mathbf{d}^* \notin R$. Thus, comparing the two paths, we can find the smallest integer $k \in [\ell]$, such that $\mathbf{v}_k^* \neq \mathbf{u}_{j_1^*,\ldots,j_k^*}$. We then obtain a collision for $h_{\mathbf{A}}$ at the parent node of $\mathbf{u}_{j_1^*,\ldots,j_k^*}$. The theorem then follows from Lemma 1. ☐

## 3.4 Zero-Knowledge AoK of an Accumulated Value

Our goal in this section is to construct a zero-knowledge argument system that allows prover $\mathcal{P}$ to convince verifier $\mathcal{V}$ that $\mathcal{P}$ knows a secret value that is properly accumulated into the root of the lattice-based Merkle tree described above. More formally, in our protocol, $\mathcal{P}$ convinces $\mathcal{V}$ on input $(\mathbf{A}, \mathbf{u})$ that $\mathcal{P}$ possesses a value-witness pair $(\mathbf{d}, w)$ such that $\mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u}, \mathbf{d}, w\big) = 1$. The associated relation $\mathrm{R}_{\mathrm{acc}}$ is defined as follows.

**Definition 5.**

$$\mathrm{R}_{\mathrm{acc}} = \Big\{\big((\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \{0,1\}^{nk}; \mathbf{d} \in \{0,1\}^{nk}, w \in \{0,1\}^{\ell} \times (\{0,1\}^{nk})^{\ell}\big) : \mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u}, \mathbf{d}, w\big) = 1\Big\}.$$

Before going into the details, we first introduce several supporting notations and techniques.

– We denote by $\mathsf{B}_m^{nk}$ the set of all vectors in $\{0,1\}^m$ that have Hamming weight $nk$; and by $\mathcal{S}_m$ the set of all permutations of $m$ elements.
– For $i \in \{nk, m\}$, for $b \in \{0,1\}$ and for $\mathbf{v} \in \{0,1\}^i$, we let $\mathsf{ext}(b, \mathbf{v})$ denote the vector $\mathbf{z} \in \{0,1\}^{2i}$ of the form $\mathbf{z} = \begin{pmatrix} \bar{b}\cdot\mathbf{v} \\ b\cdot\mathbf{v}\end{pmatrix}$.
– For $b \in \{0,1\}$, for $\pi \in \mathcal{S}_m$, we define the permutation $F_{b,\pi}$ that transforms $\mathbf{z} = \begin{pmatrix}\mathbf{z}_0\\ \mathbf{z}_1\end{pmatrix} \in \mathbb{Z}_q^{2m}$ consisting of 2 blocks of size $m$ into $F_{b,\pi}(\mathbf{z}) = \begin{pmatrix}\pi(\mathbf{z}_b)\\ \pi(\mathbf{z}_{\bar{b}})\end{pmatrix}$. Namely, $F_{b,\pi}$ first rearranges the blocks of $\mathbf{z}$ according to $b$ (it keeps the arrangement of blocks if $b = 0$, or swaps them if $b = 1$), then it permutes each block according to $\pi$.

Our strategy to achieve zero-knowledgeness will crucially rely on the following observation: For all $c, b \in \{0,1\}$, all $\pi, \phi \in \mathcal{S}_m$, and all $\mathbf{v}, \mathbf{w} \in \{0,1\}^m$, we have the equivalences

$$\begin{cases} \mathbf{z} = \mathsf{ext}(c, \mathbf{v}) \wedge \mathbf{v} \in \mathsf{B}_m^{nk} \iff F_{b,\pi}(\mathbf{z}) = \mathsf{Ext}(c \oplus b, \pi(\mathbf{v})) \wedge \pi(\mathbf{v}) \in \mathsf{B}_m^{nk}; \\ \mathbf{y} = \mathsf{ext}(\bar{c}, \mathbf{w}) \wedge \mathbf{w} \in \mathsf{B}_m^{nk} \iff F_{\bar{b},\pi}(\mathbf{y}) = \mathsf{Ext}(c \oplus b, \pi(\mathbf{w})) \wedge \pi(\mathbf{w}) \in \mathsf{B}_m^{nk}. \end{cases} \quad (1)$$

**Warm-up step.** Now, let $(\mathbf{d}, w)$ be such that $\big((\mathbf{A}, \mathbf{u}), \mathbf{d}, w\big) \in \mathrm{R}_{\mathrm{acc}}$, where $w$ is of the form $w = \big((j_1, \ldots, j_\ell), (\mathbf{w}_\ell, \ldots, \mathbf{w}_1)\big)$, and let $\mathbf{v}_\ell = \mathbf{d}, \mathbf{v}_{\ell-1}, \ldots, \mathbf{v}_1, \mathbf{v}_0$ be the path computed by $\mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u}, \mathbf{d}, w\big)$. Note that $\mathbf{v}_0 = \mathbf{u}$ and:

$$\forall i \in \{\ell-1, \ldots, 1, 0\} : \ \mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases} \quad (2)$$

We observe that relation (2) can be equivalently rewritten in a more compact form: $\forall i \in \{\ell-1, \ldots, 1, 0\}$,

$$\mathbf{v}_i = \bar{j}_{i+1} \cdot h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}) + j_{i+1} \cdot h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}). \quad (3)$$

Equation (3) then can be interpreted as:

$$\begin{aligned} &\bar{j}_{i+1} \cdot \big(\mathbf{A}_0 \cdot \mathbf{v}_{i+1} + \mathbf{A}_1 \cdot \mathbf{w}_{i+1}\big) + j_{i+1} \cdot \big(\mathbf{A}_0 \cdot \mathbf{w}_{i+1} + \mathbf{A}_1 \cdot \mathbf{v}_{i+1}\big) = \mathbf{G} \cdot \mathbf{v}_i \bmod q \\ \Leftrightarrow\ &\mathbf{A} \cdot \begin{pmatrix} \bar{j}_{i+1} \cdot \mathbf{v}_{i+1} \\ j_{i+1} \cdot \mathbf{v}_{i+1} \end{pmatrix} + \mathbf{A} \cdot \begin{pmatrix} j_{i+1} \cdot \mathbf{w}_{i+1} \\ \bar{j}_{i+1} \cdot \mathbf{w}_{i+1} \end{pmatrix} = \mathbf{G} \cdot \mathbf{v}_i \bmod q \\ \Leftrightarrow\ &\mathbf{A} \cdot \mathsf{ext}(j_{i+1}, \mathbf{v}_{i+1}) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}) = \mathbf{G} \cdot \mathbf{v}_i \bmod q. \end{aligned}$$

Therefore, to achieve our goal, it is necessary and sufficient to construct an argument system in which $\mathcal{P}$ convinces $\mathcal{V}$ in $\mathsf{ZK}$ that $\mathcal{P}$ knows $j_1, \ldots, j_\ell \in \{0,1\}^\ell$ and $\mathbf{v}_1, \ldots, \mathbf{v}_\ell, \mathbf{w}_1, \ldots, \mathbf{w}_\ell \in \{0,1\}^{nk}$ satisfying

$$\begin{cases} \mathbf{A} \cdot \mathsf{ext}(j_1, \mathbf{v}_1) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_1, \mathbf{w}_1) = \mathbf{G} \cdot \mathbf{u} \bmod q; \\ \forall i \in [\ell-1] : \ \mathbf{A} \cdot \mathsf{ext}(j_{i+1}, \mathbf{v}_{i+1}) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}) = \mathbf{G} \cdot \mathbf{v}_i \bmod q. \end{cases} \quad (4)$$

To this end, we develop a Stern-type protocol [67], in which we adapt the extension technique from [45]. Specifically, we perform the following extensions:

- Extend matrix $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1]$ to matrix $\mathbf{A}^* = [\mathbf{A}_0|\mathbf{0}^{n\times nk}|\mathbf{A}_1|\mathbf{0}^{n\times nk}] \in \mathbb{Z}_q^{n\times 2m}$.
- Extend matrix $\mathbf{G}$ to matrix $\mathbf{G}^* = [\mathbf{G}|\mathbf{0}^{n\times nk}] \in \mathbb{Z}_q^{n\times m}$.
- Extend $\mathbf{v}_1, \ldots, \mathbf{v}_\ell, \mathbf{w}_1, \ldots, \mathbf{w}_\ell$ into $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*, \mathbf{w}_1^*, \ldots, \mathbf{w}_\ell^* \in \mathsf{B}_m^{nk}$, respectively. This is done by appending a length-$nk$ vector of suitable Hamming weight to each of these vectors.

Let $\mathbf{z}_i = \mathsf{ext}(j_i, \mathbf{v}_i^*)$ and $\mathbf{y}_i = \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^*)$ for each $i \in [\ell]$. Note that now the conditions in (4) can be equivalently rewritten as:

$$\begin{cases} \mathbf{A}^* \cdot \mathbf{z}_1 + \mathbf{A}^* \cdot \mathbf{y}_1 = \mathbf{G} \cdot \mathbf{u} \bmod q; \\ \forall i \in [\ell-1] : \ \mathbf{A}^* \cdot \mathbf{z}_{i+1} + \mathbf{A}^* \cdot \mathbf{y}_{i+1} = \mathbf{G}^* \cdot \mathbf{v}_i^* \bmod q. \end{cases} \quad (5)$$

**The Interactive Protocol.** Having performed the above preparation and transformation steps, we now give a summary and sketch the main ideas of our interactive protocol, before formally describing it. The public parameters are $n, q, k, m, \ell$, the "powers-of-2" matrix $\mathbf{G}$ and its extension $\mathbf{G}^*$.

**Common inputs:** $(\mathbf{A}, \mathbf{u})$. Both parties extend $\mathbf{A}$ to $\mathbf{A}^*$.

**$\mathcal{P}$'s inputs:** $\big((j_1, \ldots, j_\ell), (\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*), (\mathbf{w}_1^*, \ldots, \mathbf{w}_\ell^*), (\mathbf{z}_1, \ldots, \mathbf{z}_\ell), (\mathbf{y}_1, \ldots, \mathbf{y}_\ell)\big)$.

**$\mathcal{P}$'s goal:** Prove in $\mathsf{ZK}$ that $\mathbf{v}_i^*, \mathbf{w}_i^* \in \mathsf{B}_m^{nk}$, $\mathbf{z}_i = \mathsf{ext}(j_i, \mathbf{v}_i^*)$, $\mathbf{y}_i = \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^*)$ for all $i \in [\ell]$, and that (5) holds.

To achieve its goal, $\mathcal{P}$ employs the following strategies:

1. To prove in $\mathsf{ZK}$ that $\mathbf{v}_i^*, \mathbf{w}_i^* \in \mathsf{B}_m^{nk}$ and $\mathbf{z}_i = \mathsf{ext}(j_i, \mathbf{v}_i^*)$ and $\mathbf{y}_i = \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^*)$ for all $i \in [\ell]$, the equivalences observed in (1) are exploited. Specifically, for each $i \in [\ell]$, $\mathcal{P}$ samples $\pi_i, \phi_i \xleftarrow{\$} \mathcal{S}_m$ and $b_i \xleftarrow{\$} \{0,1\}$, then it demonstrates to $\mathcal{V}$ that:
$$\begin{cases} \pi_i(\mathbf{v}_i^*) \in \mathsf{B}_m^{nk} \;\;\wedge\;\; F_{b_i,\pi_i}(\mathbf{z}_i) = \mathsf{ext}(j_i \oplus b_i, \pi_i(\mathbf{v}_i^*)); \\ \phi_i(\mathbf{w}_i^*) \in \mathsf{B}_m^{nk} \;\;\wedge\;\; F_{\bar{b}_i,\pi_i}(\mathbf{y}_i) = \mathsf{ext}(j_i \oplus b_i, \phi_i(\mathbf{w}_i^*)). \end{cases} \tag{6}$$
Seeing (6), $\mathcal{V}$ should be convinced of the facts $\mathcal{P}$ wants to prove, while learning no additional information, thanks to the randomness of $\pi_i, \phi_i$ and $b_i$.
2. To prove in $\mathsf{ZK}$ that the $\ell$ equations in (5) hold, $\mathcal{P}$ samples uniformly random masking vectors $\mathbf{r}_{\mathbf{v}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{v}}^{(\ell-1)} \xleftarrow{\$} \mathbb{Z}_q^m$; $\mathbf{r}_{\mathbf{z}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{z}}^{(\ell)}, \mathbf{r}_{\mathbf{y}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{y}}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^{2m}$, and then it shows $\mathcal{V}$ that
$$\begin{cases} \mathbf{A}^* \cdot (\mathbf{z}_1 + \mathbf{r}_{\mathbf{z}}^{(1)}) + \mathbf{A}^* \cdot (\mathbf{y}_1 + \mathbf{r}_{\mathbf{y}}^{(1)}) - \mathbf{G} \cdot \mathbf{u} = \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{z}}^{(1)} + \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{y}}^{(1)} \bmod q; \\ \forall i \in [\ell - 1] :\; \mathbf{A}^* \cdot (\mathbf{z}_{i+1} + \mathbf{r}_{\mathbf{z}}^{(i+1)}) + \mathbf{A}^* \cdot (\mathbf{y}_{i+1} + \mathbf{r}_{\mathbf{y}}^{(i+1)}) - \mathbf{G}^* \cdot (\mathbf{v}_i^* + \mathbf{r}_{\mathbf{v}}^{(i)}) \\ \qquad\qquad = \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{z}}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{y}}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{r}_{\mathbf{v}}^{(i)} \bmod q. \end{cases}$$

Let $\mathsf{COM} : \{0,1\}^* \times \{0,1\}^m \to \mathbb{Z}_q^n$ be the string commitment scheme from [40], which is statistically hiding and computationally binding if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard. The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 2.

## 3.5 Analysis of the Interactive Protocol

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

**Theorem 2.** *The given interactive protocol has perfect completeness and communication cost $\widetilde{\mathcal{O}}(\ell \cdot n)$. If $\mathsf{COM}$ is a statistically hiding and computationally binding string commitment scheme, then it is a statistical zero-knowledge argument of knowledge for the relation $\mathrm{R}_{\mathrm{acc}}$.*

**Completeness and Communication Cost.** Based on the discussion given in the previous section, it can be checked that the described protocol has perfect completeness, *i.e.*, if $\mathcal{P}$ is honest and follows the protocol, then $\mathcal{V}$ always

**1. Commitment.** $\mathcal{P}$ samples randomness $\rho_1, \rho_2, \rho_3$ for $\mathsf{COM}$ and

$$\begin{cases} b_1, \ldots, b_\ell \xleftarrow{\$} \{0,1\};\ \pi_1, \ldots, \pi_\ell, \phi_1, \ldots, \phi_\ell \xleftarrow{\$} \mathcal{S}_m; \\ \mathbf{r}_\mathbf{v}^{(1)}, \ldots, \mathbf{r}_\mathbf{v}^{(\ell-1)} \xleftarrow{\$} \mathbb{Z}_q^m;\ \mathbf{r}_\mathbf{z}^{(1)}, \ldots, \mathbf{r}_\mathbf{z}^{(\ell)}, \mathbf{r}_\mathbf{y}^{(1)}, \ldots, \mathbf{r}_\mathbf{y}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^{2m}. \end{cases}$$

It then sends $\mathcal{V}$ commitment $\mathrm{CMT} = (C_1, C_2, C_3)$, where

$$\begin{cases} C_1 = \mathsf{COM}\big(\ \{b_i; \pi_i; \phi_i\}_{i=1}^{\ell};\ \mathbf{A}^* \cdot \mathbf{r}_\mathbf{z}^{(1)} + \mathbf{A}^* \cdot \mathbf{r}_\mathbf{y}^{(1)}; \\ \qquad\qquad \{\mathbf{A}^* \cdot \mathbf{r}_\mathbf{z}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{r}_\mathbf{y}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{r}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \rho_1\big) \\ C_2 = \mathsf{COM}\big(\ \{\pi_i(\mathbf{r}_\mathbf{v}^{(i)})\}_{i=1}^{\ell-1};\ \{F_{b_i,\pi_i}(\mathbf{r}_\mathbf{z}^{(i)});\ F_{\bar{b}_i,\phi_i}(\mathbf{r}_\mathbf{y}^{(i)})\}_{i=1}^{\ell};\ \rho_2\big) \\ C_3 = \mathsf{COM}\big(\ \{\pi_i(\mathbf{v}_i^* + \mathbf{r}_\mathbf{v}^{(i)})\}_{i=1}^{\ell-1};\ \{F_{b_i,\pi_i}(\mathbf{z}_i + \mathbf{r}_\mathbf{z}^{(i)});\ F_{\bar{b}_i,\phi_i}(\mathbf{y}_i + \mathbf{r}_\mathbf{y}^{(i)})\}_{i=1}^{\ell};\ \rho_3\big). \end{cases}$$

**2. Challenge.** Receiving CMT, $\mathcal{V}$ sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$ to $\mathcal{P}$.

**3. Response.** Depending on $Ch$, $\mathcal{P}$ sends the response RSP computed as follows:

– Case $Ch = 1$: For each $i \in [\ell - 1]$, let $\mathbf{t}_\mathbf{v}^{(i)} = \pi_i(\mathbf{r}_\mathbf{v}^{(i)})$. For each $i \in [\ell]$, let:

$$a_i = j_i \oplus b_i;\ \mathbf{s}_\mathbf{v}^{(i)} = \pi_i(\mathbf{v}_i^*);\ \mathbf{s}_\mathbf{w}^{(i)} = \phi_i(\mathbf{w}_i^*);\ \mathbf{t}_\mathbf{z}^{(i)} = F_{b_i,\pi_i}(\mathbf{r}_\mathbf{z}^{(i)});\ \mathbf{t}_\mathbf{y}^{(i)} = F_{\bar{b}_i,\phi_i}(\mathbf{r}_\mathbf{y}^{(i)}).$$

$$\text{Then let RSP} = \big(\ \{\mathbf{t}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \{a_i;\ \mathbf{s}_\mathbf{v}^{(i)};\ \mathbf{t}_\mathbf{z}^{(i)};\ \mathbf{s}_\mathbf{w}^{(i)};\ \mathbf{t}_\mathbf{y}^{(i)}\}_{i=1}^{\ell};\ \rho_2; \rho_3\big). \tag{7}$$

– Case $Ch = 2$: For each $i \in [\ell - 1]$, let $\mathbf{e}_\mathbf{v}^{(i)} = \mathbf{v}_i^* + \mathbf{r}_\mathbf{v}^{(i)}$. For each $i \in [\ell]$, let:

$$c_i = b_i;\ \widehat{\pi}_i = \pi_i;\ \widehat{\phi}_i = \phi_i;\ \mathbf{e}_\mathbf{z}^{(i)} = \mathbf{z}_i + \mathbf{r}_\mathbf{z}^{(i)};\ \mathbf{e}_\mathbf{y}^{(i)} = \mathbf{y}_i + \mathbf{r}_\mathbf{y}^{(i)}.$$

$$\text{Then let RSP} = \big(\ \{\mathbf{e}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \{c_i;\ \widehat{\pi}_i;\ \widehat{\phi}_i;\ \mathbf{e}_\mathbf{z}^{(i)};\ \mathbf{e}_\mathbf{y}^{(i)}\}_{i=1}^{\ell};\ \rho_1; \rho_3\big). \tag{8}$$

– Case $Ch = 3$: For each $i \in [\ell - 1]$, let $\mathbf{p}_\mathbf{v}^{(i)} = \mathbf{r}_\mathbf{v}^{(i)}$. For each $i \in [\ell]$, let:

$$d_i = b_i;\ \widetilde{\pi}_i = \pi_i;\ \widetilde{\phi}_i = \phi_i;\ \mathbf{p}_\mathbf{z}^{(i)} = \mathbf{r}_\mathbf{z}^{(i)};\ \mathbf{p}_\mathbf{y}^{(i)} = \mathbf{r}_\mathbf{y}^{(i)}.$$

$$\text{Then let RSP} = \big(\ \{\mathbf{p}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \{d_i;\ \widetilde{\pi}_i;\ \widetilde{\phi}_i;\ \mathbf{p}_\mathbf{z}^{(i)};\ \mathbf{p}_\mathbf{y}^{(i)}\}_{i=1}^{\ell};\ \rho_1; \rho_2\big). \tag{9}$$

**Verification.** Receiving RSP, $\mathcal{V}$ proceeds as follows.

– Case $Ch = 1$: Parse RSP as in (7). Check that $\mathbf{s}_\mathbf{v}^{(i)}, \mathbf{s}_\mathbf{w}^{(i)} \in \mathsf{B}_m^{nk}$ for all $i \in [\ell]$. Next, for each $i \in [\ell]$, let $\mathbf{s}_\mathbf{z}^{(i)} = \mathsf{ext}(a_i, \mathbf{s}_\mathbf{v}^{(i)})$ and let $\mathbf{s}_\mathbf{y}^{(i)} = \mathsf{ext}(a_i, \mathbf{s}_\mathbf{w}^{(i)})$. Then check that:

$$\begin{cases} C_2 = \mathsf{COM}\big(\ \{\mathbf{t}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \{\mathbf{t}_\mathbf{z}^{(i)};\ \mathbf{t}_\mathbf{y}^{(i)}\}_{i=1}^{\ell};\ \rho_2\big), \\ C_3 = \mathsf{COM}\big(\ \{\mathbf{s}_\mathbf{v}^{(i)} + \mathbf{t}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \{\mathbf{s}_\mathbf{z}^{(i)} + \mathbf{t}_\mathbf{z}^{(i)};\ \mathbf{s}_\mathbf{y}^{(i)} + \mathbf{t}_\mathbf{y}^{(i)}\}_{i=1}^{\ell};\ \rho_3\big). \end{cases} \tag{10}$$

– Case $Ch = 2$: Parse RSP as in (8) and check that:

$$\begin{cases} C_1 = \mathsf{COM}\big(\ \{c_i; \widehat{\pi}_i; \widehat{\phi}_i\}_{i=1}^{\ell};\ \mathbf{A}^* \cdot \mathbf{e}_\mathbf{z}^{(1)} + \mathbf{A}^* \cdot \mathbf{e}_\mathbf{y}^{(1)} - \mathbf{G} \cdot \mathbf{u}; \\ \qquad\qquad \{\mathbf{A}^* \cdot \mathbf{e}_\mathbf{z}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{e}_\mathbf{y}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{e}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \rho_1\big) \\ C_3 = \mathsf{COM}\big(\ \{\widehat{\pi}_i(\mathbf{e}_\mathbf{v}^{(i)})\}_{i=1}^{\ell-1};\ \{F_{c_i,\widehat{\pi}_i}(\mathbf{e}_\mathbf{z}^{(i)});\ F_{\bar{c}_i,\widehat{\phi}_i}(\mathbf{e}_\mathbf{y}^{(i)})\}_{i=1}^{\ell};\ \rho_3\big). \end{cases} \tag{11}$$

– Case $Ch = 3$: Parse RSP as in (9) and check that:

$$\begin{cases} C_1 = \mathsf{COM}\big(\ \{d_i; \widetilde{\pi}_i; \widetilde{\phi}_i\}_{i=1}^{\ell};\ \mathbf{A}^* \cdot \mathbf{p}_\mathbf{z}^{(1)} + \mathbf{A}^* \cdot \mathbf{p}_\mathbf{y}^{(1)}; \\ \qquad\qquad \{\mathbf{A}^* \cdot \mathbf{p}_\mathbf{z}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{p}_\mathbf{y}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{p}_\mathbf{v}^{(i)}\}_{i=1}^{\ell-1};\ \rho_1\big) \\ C_2 = \mathsf{COM}\big(\ \{\widetilde{\pi}_i(\mathbf{p}_\mathbf{v}^{(i)})\}_{i=1}^{\ell-1};\ \{F_{d_i,\widetilde{\pi}_i}(\mathbf{p}_\mathbf{z}^{(i)});\ F_{\bar{d}_i,\widetilde{\phi}_i}(\mathbf{p}_\mathbf{y}^{(i)})\}_{i=1}^{\ell};\ \rho_2\big). \end{cases} \tag{12}$$

In each case, $\mathcal{V}$ outputs 1 if all the conditions hold. Otherwise, it outputs 0.

**Fig. 2:** A zero-knowledge argument of knowledge for the relation $\mathrm{R}_{\mathrm{acc}}$.

outputs 1. It can also be seen that the communication cost of the protocol is $\widetilde{\mathcal{O}}(\ell \cdot m \cdot \log q) = \widetilde{\mathcal{O}}(\ell \cdot n)$ bits.

In order to prove that the protocol is a $\mathsf{ZKAoK}$ for the relation $\mathrm{R}_{\mathrm{acc}}$, we will employ the standard simulation and extraction techniques for Stern-type protocols (see, *e.g.*, [40,45,46]).

**Lemma 2 (Zero-Knowledge Property).** *If* $\mathsf{COM}$ *is statistically hiding, then the interactive protocol in Figure 2 is a statistical zero-knowledge argument.*

*Proof.* We construct a PPT simulator $\mathcal{S}$ interacting with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$, such that, given only the public input, $\mathcal{S}$ 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 $\mathcal{S}$ begins by selecting a random $\overline{Ch} \in \{1, 2, 3\}$. This is a prediction of the challenge value that $\widehat{\mathcal{V}}$ will *not* choose.

**Case** $\overline{Ch} = 1$: Using linear algebra, $\mathcal{S}$ computes $\mathbf{z}'_1, \ldots, \mathbf{z}'_\ell, \mathbf{y}'_1, \ldots, \mathbf{y}'_\ell \in \mathbb{Z}_q^{2m}$ and $\mathbf{v}'_1, \ldots, \mathbf{v}'_{\ell-1} \in \mathbb{Z}_q^m$ such that

$$\begin{cases} \mathbf{A}^* \cdot \mathbf{z}'_1 + \mathbf{A}^* \cdot \mathbf{y}'_1 = \mathbf{G} \cdot \mathbf{u} \bmod q; \\ \forall i \in [1, \ell-1]: \ \mathbf{A}^* \cdot \mathbf{z}'_{i+1} + \mathbf{A}^* \cdot \mathbf{y}'_{i+1} = \mathbf{G}^* \cdot \mathbf{v}'_i \bmod q. \end{cases}$$

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

$$\begin{cases} b_1, \ldots, b_\ell \xleftarrow{\$} \{0,1\}; \ \pi_1, \ldots, \pi_\ell, \phi_1, \ldots, \phi_\ell \xleftarrow{\$} \mathcal{S}_m; \\ \mathbf{r}_{\mathbf{v}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{v}}^{(\ell-1)} \xleftarrow{\$} \mathbb{Z}_q^m; \ \mathbf{r}_{\mathbf{z}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{z}}^{(\ell)}, \mathbf{r}_{\mathbf{y}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{y}}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^{2m}. \end{cases}$$

It then sends $\widehat{\mathcal{V}}$ commitment $\mathrm{CMT} = (C'_1, C'_2, C'_3)$, where

$$\begin{cases} C'_1 = \mathsf{COM}\big(\ \{b_i;\, \pi_i;\, \phi_i\}_{i=1}^{\ell};\ \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{z}}^{(1)} + \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{y}}^{(1)}; \\ \qquad\qquad \{\mathbf{A}^* \cdot \mathbf{r}_{\mathbf{z}}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{r}_{\mathbf{y}}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{r}_{\mathbf{v}}^{(i)}\}_{i=1}^{\ell-1};\ \rho_1\big) \\ C'_2 = \mathsf{COM}\big(\ \{\pi_i(\mathbf{r}_{\mathbf{v}}^{(i)})\}_{i=1}^{\ell-1};\ \{F_{b_i,\pi_i}(\mathbf{r}_{\mathbf{z}}^{(i)});\ F_{\bar{b}_i,\phi_i}(\mathbf{r}_{\mathbf{y}}^{(i)})\}_{i=1}^{\ell};\ \rho_2\big) \\ C'_3 = \mathsf{COM}\big(\ \{\pi_i(\mathbf{v}'_i + \mathbf{r}_{\mathbf{v}}^{(i)})\}_{i=1}^{\ell-1};\ \{F_{b_i,\pi_i}(\mathbf{z}'_i + \mathbf{r}_{\mathbf{z}}^{(i)});\ F_{\bar{b}_i,\phi_i}(\mathbf{y}'_i + \mathbf{r}_{\mathbf{y}}^{(i)})\}_{i=1}^{\ell};\ \rho_3\big). \end{cases} \tag{13}$$

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

- If $Ch = 1$: Output $\perp$ and abort.
- If $Ch = 2$: Send $\mathrm{RSP} = \big(\ \{\mathbf{v}'_i + \mathbf{r}_{\mathbf{v}}^{(i)}\}_{i=1}^{\ell-1};\ \{b_i;\ \pi_i;\ \phi_i;\ \mathbf{z}'_i + \mathbf{r}_{\mathbf{z}}^{(i)};\ \mathbf{y}'_i + \mathbf{r}_{\mathbf{y}}^{(i)}\}_{i=1}^{\ell};\ \rho_1; \rho_3\big)$.
- If $Ch = 3$: Send $\mathrm{RSP} = \big(\ \{\mathbf{r}_{\mathbf{v}}^{(i)}\}_{i=1}^{\ell-1};\ \{b_i;\ \pi_i;\ \phi_i;\ \mathbf{r}_{\mathbf{z}}^{(i)};\ \mathbf{r}_{\mathbf{y}}^{(i)}\}_{i=1}^{\ell};\ \rho_1; \rho_2\big)$.

**Case** $\overline{Ch} = 2$: $\mathcal{S}$ samples

$$\begin{cases} j'_1, \ldots, j'_\ell \xleftarrow{\$} \{0,1\}; \ \mathbf{v}'_1, \ldots, \mathbf{v}'_\ell, \mathbf{w}'_1, \ldots, \mathbf{w}'_\ell \xleftarrow{\$} \mathsf{B}_m^{nk}; \\ b_1, \ldots, b_\ell \xleftarrow{\$} \{0,1\}; \ \pi_1, \ldots, \pi_\ell, \phi_1, \ldots, \phi_\ell \xleftarrow{\$} \mathcal{S}_m; \\ \mathbf{r}_{\mathbf{v}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{v}}^{(\ell-1)} \xleftarrow{\$} \mathbb{Z}_q^m; \ \mathbf{r}_{\mathbf{z}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{z}}^{(\ell)}, \mathbf{r}_{\mathbf{y}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{y}}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^{2m}. \end{cases}$$

It then computes $\mathbf{z}'_i = \mathsf{ext}(j'_i, \mathbf{v}'_i)$, $\mathbf{y}'_i = \mathsf{ext}(\bar{j}'_i, \mathbf{w}'_i)$ for each $i \in [\ell]$, and sends the commitment CMT computed in the same manner as in (13).

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

– If $Ch = 1$: Send

$$\text{RSP} = \big(\{\pi_i(\mathbf{r}_{\mathbf{v}}^{(i)})\}_{i=1}^{\ell-1};\ \{j'_i \oplus b_i;\ \pi_i(\mathbf{v}'_i);\ F_{b_i,\pi_i}(\mathbf{r}_{\mathbf{z}}^{(i)});\ \phi_i(\mathbf{w}'_i);\ F_{\bar{b}_i,\phi_i}(\mathbf{r}_{\mathbf{y}}^{(i)})\}_{i=1}^{\ell};\ \rho_2; \rho_3\big).$$

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

**Case** $\overline{Ch} = 3$: The simulator proceeds with the preparation as in the case $\overline{Ch} = 2$ above. Then it sends the commitment CMT $:= (C'_1, C'_2, C'_3)$, where $C'_2, C'_3$ are computed as in (13), while

$$C'_1 = \mathsf{COM}\big(\ \{b_i; \pi_i; \phi_i\}_{i=1}^{\ell};\ \mathbf{A}^* \cdot (\mathbf{z}'_1 + \mathbf{r}_{\mathbf{z}}^{(1)}) + \mathbf{A}^* \cdot (\mathbf{y}'_1 + \mathbf{r}_{\mathbf{y}}^{(1)}) - \mathbf{G} \cdot \mathbf{u};$$
$$\{\mathbf{A}^* \cdot (\mathbf{z}'_{i+1} + \mathbf{r}_{\mathbf{z}}^{(i+1)}) + \mathbf{A}^* \cdot (\mathbf{y}'_{i+1} + \mathbf{r}_{\mathbf{y}}^{(i+1)}) - \mathbf{G}^* \cdot (\mathbf{v}'_i + \mathbf{r}_{\mathbf{v}}^{(i)})\}_{i=1}^{\ell-1};\ \rho_1\big).$$

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

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

We observe that, in every case we have considered above, since $\mathsf{COM}$ is statistically hiding, the distribution of the commitment CMT and the distribution of the challenge $Ch$ from $\widehat{\mathcal{V}}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly 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. □

To prove that our protocol is an argument of knowledge for the relation $\mathrm{R}_{\mathrm{acc}}$, it suffices to demonstrate that the protocol has the special soundness property [34].

**Lemma 3 (Argument of Knowledge Property).** *If* $\mathsf{COM}$ *is computationally binding, then there exists an efficient knowledge extractor* $\mathcal{K}$ *that, on input* 3 *valid responses* $(\text{RSP}_1, \text{RSP}_2, \text{RSP}_3)$ *to the same commitment* CMT, *outputs a pair* $(\mathbf{d}' \in \{0,1\}^{nk}, w' \in \{0,1\}^{\ell} \times (\{0,1\}^{nk})^{\ell})$ *such that*

$$\big((\mathbf{A}, \mathbf{u}); \mathbf{d}', w'\big) \in \mathrm{R}_{\mathrm{acc}}.$$

*Proof.* Let the 3 valid responses to CMT $= (C_1, C_2, C_3)$ be

$$\begin{cases} \text{RSP}_1 = \big(\ \{\mathbf{t}_{\mathbf{v}}^{(i)}\}_{i=1}^{\ell-1};\ \{a_i;\ \mathbf{s}_{\mathbf{v}}^{(i)};\ \mathbf{t}_{\mathbf{z}}^{(i)};\ \mathbf{s}_{\mathbf{w}}^{(i)};\ \mathbf{t}_{\mathbf{y}}^{(i)}\}_{i=1}^{\ell};\ \rho_2; \rho_3\big), \\ \text{RSP}_2 = \big(\ \{\mathbf{e}_{\mathbf{v}}^{(i)}\}_{i=1}^{\ell-1};\ \{c_i;\ \widehat{\pi}_i;\ \widehat{\phi}_i;\ \mathbf{e}_{\mathbf{z}}^{(i)};\ \mathbf{e}_{\mathbf{y}}^{(i)}\}_{i=1}^{\ell};\ \rho_1; \rho_3\big), \\ \text{RSP}_3 = \big(\ \{\mathbf{p}_{\mathbf{v}}^{(i)}\}_{i=1}^{\ell-1};\ \{d_i;\ \widetilde{\pi}_i;\ \widetilde{\phi}_i;\ \mathbf{p}_{\mathbf{z}}^{(i)};\ \mathbf{p}_{\mathbf{y}}^{(i)}\}_{i=1}^{\ell};\ \rho_1; \rho_2\big). \end{cases}$$

The validity of $\mathrm{RSP}_1$ implies that $\forall i \in [\ell] : \mathbf{s}_{\mathbf{v}}^{(i)}, \mathbf{s}_{\mathbf{w}}^{(i)} \in \mathsf{B}_m^{nk}$. Furthermore, it follows from the verification conditions given in (10), (11), (12), and from the computational binding property of $\mathsf{COM}$ that:

$$\mathbf{A}^* \cdot \mathbf{e}_{\mathbf{z}}^{(1)} + \mathbf{A}^* \cdot \mathbf{e}_{\mathbf{y}}^{(1)} - \mathbf{G} \cdot \mathbf{u} = \mathbf{A}^* \cdot \mathbf{p}_{\mathbf{z}}^{(1)} + \mathbf{A}^* \cdot \mathbf{p}_{\mathbf{y}}^{(1)} \bmod q,$$

and for all $i \in [1, \ell - 1]$: $\mathbf{t}_{\mathbf{v}}^{(i)} = \widetilde{\pi}_i(\mathbf{p}_{\mathbf{v}}^{(i)})$; $\mathbf{s}_{\mathbf{v}}^{(i)} + \mathbf{t}_{\mathbf{v}}^{(i)} = \widetilde{\pi}_i(\mathbf{e}_{\mathbf{v}}^{(i)})$; and

$$\mathbf{A}^* \cdot \mathbf{e}_{\mathbf{z}}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{e}_{\mathbf{y}}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{e}_{\mathbf{v}}^{(i)} = \mathbf{A}^* \cdot \mathbf{p}_{\mathbf{z}}^{(i+1)} + \mathbf{A}^* \cdot \mathbf{p}_{\mathbf{y}}^{(i+1)} - \mathbf{G}^* \cdot \mathbf{p}_{\mathbf{v}}^{(i)} \bmod q,$$

and for all $i \in [\ell]$:

$$\begin{cases} c_i = d_i;\ \widehat{\pi}_i = \widetilde{\pi}_i;\ \widehat{\phi}_i = \widetilde{\phi}_i; \\ \mathbf{t}_{\mathbf{z}}^{(i)} = F_{d_i, \widetilde{\pi}_i}(\mathbf{p}_{\mathbf{z}}^{(i)});\ \mathsf{ext}(a_i, \mathbf{s}_{\mathbf{v}}^{(i)}) + \mathbf{t}_{\mathbf{z}}^{(i)} = F_{c_i, \widehat{\pi}_i}(\mathbf{e}_{\mathbf{z}}^{(i)}); \\ \mathbf{t}_{\mathbf{y}}^{(i)} = F_{\bar{d}_i, \widetilde{\phi}_i}(\mathbf{p}_{\mathbf{y}}^{(i)});\ \mathsf{ext}(a_i, \mathbf{s}_{\mathbf{w}}^{(i)}) + \mathbf{t}_{\mathbf{y}}^{(i)} = F_{\bar{c}_i, \widehat{\phi}_i}(\mathbf{e}_{\mathbf{y}}^{(i)}). \end{cases}$$

The knowledge extractor $\mathcal{K}$ now proceeds as follows. For each $i \in [\ell]$, let:

$$j_i = a_i \oplus c_i;\ \mathbf{v}_i^* = \widehat{\pi}_i^{-1}(\mathbf{s}_{\mathbf{v}}^{(i)});\ \mathbf{w}_i^* = \widehat{\phi}_i^{-1}(\mathbf{s}_{\mathbf{w}}^{(i)});\ \mathbf{z}_i = \mathbf{e}_{\mathbf{z}}^{(i)} - \mathbf{p}_{\mathbf{z}}^{(i)};\ \mathbf{y}_i = \mathbf{e}_{\mathbf{y}}^{(i)} - \mathbf{p}_{\mathbf{y}}^{(i)}.$$

Note that $\widehat{\pi}_i(\mathbf{v}_i^*) = \mathbf{s}_{\mathbf{v}}^{(i)} \in \mathsf{B}_m^{nk}$, and thus $\mathbf{v}_i^* \in \mathsf{B}_m^{nk}$ (by (1)). Similarly, $\mathbf{w}_i^* \in \mathsf{B}_m^{nk}$. Furthermore, one has that:

- $F_{c_i, \widehat{\pi}_i}(\mathbf{z}_i) = \mathsf{ext}(a_i, \mathbf{s}_{\mathbf{v}}^{(i)}) = \mathsf{ext}\big(j_i \oplus c_i, \widehat{\pi}_i(\mathbf{v}_i^*)\big)$. By (1), this implies $\mathbf{z}_i = \mathsf{ext}(j_i, \mathbf{v}_i^*)$.
- $F_{\bar{c}_i, \widehat{\phi}_i}(\mathbf{y}_i) = \mathsf{ext}(a_i, \mathbf{s}_{\mathbf{w}}^{(i)}) = \mathsf{ext}\big(\bar{j}_i \oplus \bar{c}_i, \widehat{\phi}_i(\mathbf{w}_i^*)\big)$. By (1), this implies $\mathbf{y}_i = \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^*)$.

Moreover, the following relations hold:

$$\begin{cases} \mathbf{A}^* \cdot \mathbf{z}_1 + \mathbf{A}^* \cdot \mathbf{y}_1 = \mathbf{G} \cdot \mathbf{u} \bmod q \\ \forall i \in [1, \ell-1] :\ \mathbf{A}^* \cdot \mathbf{z}_{i+1} + \mathbf{A}^* \cdot \mathbf{y}_{i+1} = \mathbf{G}^* \cdot \mathbf{v}_i^* \bmod q \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{A}^* \cdot \mathsf{ext}(j_1, \mathbf{v}_1^*) + \mathbf{A}^* \cdot \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^*) = \mathbf{G} \cdot \mathbf{u} \bmod q \\ \forall i \in [1, \ell-1] :\ \mathbf{A}^* \cdot \mathsf{ext}(j_{i+1}, \mathbf{v}_{i+1}^*) + \mathbf{A}^* \cdot \mathsf{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}^*) = \mathbf{G}^* \cdot \mathbf{v}_i^* \bmod q. \end{cases}$$

Now, by dropping the last $nk$ coordinates from $\mathbf{v}_1^*, \ldots, \mathbf{v}_\ell^*, \mathbf{w}_1^*, \ldots, \mathbf{w}_\ell^*$, the knowledge extractor $\mathcal{K}$ obtains $\mathbf{v}'_1, \ldots, \mathbf{v}'_\ell, \mathbf{w}'_1, \ldots, \mathbf{w}'_\ell \in \{0,1\}^{nk}$, respectively. These vectors satisfy:

$$\begin{cases} \mathbf{A} \cdot \mathsf{ext}(j_1, \mathbf{v}'_1) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_1, \mathbf{w}'_1) = \mathbf{G} \cdot \mathbf{u} \bmod q \\ \forall i \in [1, \ell-1] :\ \mathbf{A} \cdot \mathsf{ext}(j_{i+1}, \mathbf{v}'_{i+1}) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_{i+1}, \mathbf{w}'_{i+1}) = \mathbf{G} \cdot \mathbf{v}'_i \bmod q \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{v}'_0 = \mathbf{u} \\ \forall i \in [0, \ell-1] :\ \mathbf{v}'_i = \bar{j}_{i+1} \cdot h_{\mathbf{A}}(\mathbf{v}'_{i+1}, \mathbf{w}'_{i+1}) + j_{i+1} \cdot h_{\mathbf{A}}(\mathbf{w}'_{i+1}, \mathbf{v}'_{i+1}). \end{cases}$$

Let $\mathbf{d}' = \mathbf{v}'_\ell$ and $w' = \big((j_1, \ldots, j_\ell), (\mathbf{w}'_\ell, \ldots, \mathbf{w}'_1)\big)$, then $\mathsf{TVerify}_{\mathbf{A}}(\mathbf{u}, \mathbf{d}', w') = 1$. In other words, $(\mathbf{d}', w')$ satisfies $\big((\mathbf{A}, \mathbf{u}); \mathbf{d}', w'\big) \in \mathrm{R}_{\mathrm{acc}}$. This concludes the proof. $\square$

## 4 A Logarithmic-Size Ring Signature from Lattices

In this section, we construct a ring signature scheme [65] with signature size $\widetilde{\mathcal{O}}(\log N \cdot n)$, where $N$ is the size of the ring, based on the hardness of lattice problem $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$. We use the $\mathsf{ZKAoK}$ given in Section 3 as the building block.

### 4.1 Definitions

We recall the standard definitions and security requirements for ring signatures [11,36]. A ring signature scheme consists of a tuple of efficient algorithms $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify})$ for generating a public parameter, generating keys for users, signing messages, and verifying ring signatures, respectively.

**$\mathsf{RSetup}(n)$:** Generates public parameters $pp$ which are made available to all users.
**$\mathsf{RKgen}(pp)$:** Generates a public key $pk$ and the corresponding secret key $sk$.
**$\mathsf{RSign}_{pp}(sk, M, R)$:** Outputs a signature $\Sigma$ on the message $M \in \{0,1\}^*$ with respect to the ring $R = (pk_0, \ldots, pk_{N-1})$. It is required that $(pk, sk)$ be a valid key pair produced by $\mathsf{RKgen}(pp)$ and that $pk \in R$.
**$\mathsf{RVerify}_{pp}(M, R, \Sigma)$:** Given a candidate signature $\Sigma$ on a message $M$ with respect to the ring of public keys $R$, this algorithm outputs 1 if $\Sigma$ is deemed valid or 0 otherwise.

We next describe the following requirements for ring signatures: correctness, unforgeability with respect to insider corruption, and statistical anonymity.

The correctness requirement says that a user can always sign any message on behalf of a ring he belongs to. This is formalized as follows.

**Definition 6 (Correctness).** A ring signature $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify})$ is correct if for any $pp \leftarrow \mathsf{RSetup}(n)$, any $(pk, sk) \leftarrow \mathsf{RKgen}(pp)$, any $R$ such that $pk \in R$, any $M \in \{0,1\}^*$, we have $\mathsf{RVerify}_{pp}\big(M, R, \mathsf{RSign}_{pp}(sk, M, R)\big) = 1$.

A ring signature is unforgeable with respect to insider corruption if it is infeasible to forge a ring signature without controlling one of the ring members.

**Definition 7 (Unforgeability w.r.t. insider corruption).** *A ring signature scheme* $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify})$ *is unforgeable w.r.t. insider corruption if for all PPT adversaries* $\mathcal{A}$*,*

$$\begin{aligned}\Pr[pp \leftarrow \mathsf{RSetup}(1^n); (M^\star, R^\star, \Sigma^\star) \leftarrow \mathcal{A}^{\mathrm{PKGen},\mathrm{Sign},\mathrm{Corrupt}}(pp) :& \\ \mathsf{RVerify}_{pp}(M^\star, R^\star, \Sigma^\star) = 1] \in \mathsf{negl}(n),&\end{aligned}$$

*where:*

- PKGen *on the $j$-th query runs* $(pk_j, sk_j) \leftarrow \mathsf{RKgen}(pp)$ *and returns* $pk_j$*.*
- $\mathrm{Sign}(j, M, R)$ *returns the output of* $\mathsf{RSign}_{pp}(sk_j, M, R)$ *provided: (i)* $(pk_j, sk_j)$ *has been generated by* PKGen*; (ii)* $pk_j \in R$*. Otherwise, it returns* $\perp$*.*
- $\mathrm{Corrupt}(j)$ *returns* $sk_j$*, provided that* $(pk_j, sk_j)$ *has been generated by* PKGen*.*

– *$\mathcal{A}$ outputs $(M^\star, R^\star, \Sigma^\star)$ such that* Sign$(\cdot, M^\star, R^\star)$ *has not been queried. Moreover, $R^\star$ is non-empty and only contains public keys $pk_j$ generated by* PKGen *for which $j$ has not been corrupted.*

**Definition 8.** A ring signature scheme $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify})$ provides statistical anonymity if, for any (possibly unbounded) adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{RSetup}(1^n); (M^\star, j_0, j_1, R^\star) \leftarrow \mathcal{A}^{\mathsf{RKgen}(pp)}(pp) \\ b \xleftarrow{\$} \{0,1\}; \Sigma^* \leftarrow \mathsf{RSign}_{pp}(sk_{j_b}, M^\star, R^\star) \end{array} : \mathcal{A}(\Sigma^\star) = b\right]$$
$$= 1/2 + \mathsf{negl}(n),$$

*where $pk_{j_0}, pk_{j_1} \in R^\star$.*

*Remark:* Anonymity under full key exposure [11] requires that the randomness used by KeyGen be revealed to the adversary. In our construction, it does not make a difference since we assume computationally unbounded adversaries. A $c$-user ring signature scheme is a variant of ring signatures, that only supports rings of fixed size $c$. Here, we do not assume any upper bound on the size of a ring. Similarly to [36], we only assume that all users agree on pre-existing public parameters $pp$. In our scheme, these public parameters consist of a modulus $q$ and a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times 2nk}$ which can be derived from a random oracle. In this case, we only need all users to agree on the parameters $q$ and $n$.

## 4.2 The Underlying Zero-Knowledge Protocol

The ring signature scheme that we will present next relies on a simple extension of the $\mathsf{ZKAoK}$ in Section 3. Specifically, one more layer is added: apart from proving that it has a secret value $\mathbf{d}$ that was properly accumulated to the root of the tree, $\mathcal{P}$ has to convince $\mathcal{V}$ that it knows a vector $\mathbf{x} \in \{0,1\}^m$ such that $\mathsf{bin}(\mathbf{A}\cdot\mathbf{x} \bmod q) = \mathbf{d}$, or equivalently, $\mathbf{A}\cdot\mathbf{x} = \mathbf{G}\cdot\mathbf{d} \bmod q$. The associated relation $\mathrm{R}_{\mathrm{ring}}$ is defined as follows.

**Definition 9.** *Define the relation*

$$\mathrm{R}_{\mathrm{ring}} = \Big\{\big((\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \{0,1\}^{nk}; \mathbf{d} \in \{0,1\}^{nk}, w \in \{0,1\}^{\ell} \times (\{0,1\}^{nk})^{\ell},$$
$$\mathbf{x} \in \{0,1\}^m\big) : \mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u}, \mathbf{d}, w\big) = 1 \wedge \mathbf{A}\cdot\mathbf{x} = \mathbf{G}\cdot\mathbf{d} \bmod q\Big\}.$$

A $\mathsf{ZKAoK}$ for $\mathrm{R}_{\mathrm{ring}}$ can be obtained from the one in Section 3, where the new layer is handled by the same “extend-then-permute” technique. As before, the protocol relies on the string commitment scheme from [40], which is statistically hiding and computationally binding if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard.

**Lemma 4.** *Let us assume that the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard. Then, there exists a statistical $\mathsf{ZKAoK}$ for the relation $\mathrm{R}_{\mathrm{ring}}$ with perfect completeness and communication cost $\widetilde{\mathcal{O}}(\ell \cdot n)$. In particular:*

– *There exists an efficient simulator that, on input* $(\mathbf{A}, \mathbf{u})$*, outputs an accepted transcript which is statistically close to that produced by the real prover.*
– *There exists an efficient knowledge extractor that, on input* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to the same commitment* CMT*, outputs* $(\mathbf{d}', w', \mathbf{x}')$ *such that*

$$\big((\mathbf{A}, \mathbf{u}), \mathbf{d}', w', \mathbf{x}'\big) \in \mathrm{R}_{\mathrm{ring}}.$$

The full description and analysis of the argument system are given in the full version of the paper.

### 4.3 Description of the Ring Signature Scheme

We now will construct a ring signature scheme for rings of $N = 2^\ell$ users based on the Merkle-tree accumulator presented in Section 3. Our ring signature can be easily adapted for the case when the size of the ring is not a power of 2 (see Remark 1). The scheme uses parameters $n, m, q$ defined as in Section 3, parameter $\kappa = \omega(\log n)$ that determines the number of protocol repetitions, and a random oracle $\mathcal{H}_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^\kappa$.

$\mathsf{RSetup}(n)$ : Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and output $pp = \mathbf{A}$.

$\mathsf{RKgen}(pp = \mathbf{A})$ : Pick $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$, compute $\mathbf{d} = \mathsf{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q) \in \{0,1\}^{nk}$, and output $(sk, pk) = (\mathbf{x}, \mathbf{d})$.

$\mathsf{RSign}_{pp}(sk, M, R)$ : Given a ring $R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1})$, where $\mathbf{d}_i \in \{0,1\}^{nk}$ for every $i \in [0, N-1]$, and $sk = \mathbf{x} \in \{0,1\}^m$ such that $\mathbf{d} = \mathsf{bin}(\mathbf{A}\mathbf{x} \bmod q) \in R$, this algorithm generates a ring signature $\Sigma$ on $M \in \{0,1\}^*$ as follows:

1. Run algorithm $\mathsf{TAcc}_{\mathbf{A}}(R)$ to build the Merkle tree based on $R$ and the hash function $h_{\mathbf{A}}$, and obtain the root $\mathbf{u} \in \{0,1\}^{nk}$.
2. Run algorithm $\mathsf{TWitness}_{\mathbf{A}}(R, \mathbf{d})$ to get a witness

$$w = \big((j_1, \ldots, j_\ell) \in \{0,1\}^\ell, (\mathbf{w}_\ell, \ldots, \mathbf{w}_1) \in (\{0,1\}^{nk})^\ell\big)$$

   to the fact that $\mathbf{d}$ was properly accumulated in $\mathbf{u}$.
3. Generate a $\mathsf{NIZKAoK}$ $\Pi_{\mathsf{ring}}$ to demonstrate the possession of a valid pair $(sk, pk) = (\mathbf{x}, \mathbf{d})$ such that $\mathbf{d}$ is properly accumulated in $\mathbf{u}$. This is done by running the protocol in Section 4.2 with public input $(\mathbf{A}, \mathbf{u})$ and prover's witness $(\mathbf{x}, \mathbf{d}, w)$. The protocol is repeated $\kappa = \omega(\log n)$ times to achieve negligible soundness error and made non-interactive via the Fiat-Shamir heuristic as a triple $\Pi_{\mathsf{ring}} = (\{\mathrm{CMT}_i\}_{i=1}^\kappa, \mathrm{CH}, \{\mathrm{RSP}\}_{i=1}^\kappa)$, where

$$\mathrm{CH} = \mathcal{H}_{\mathsf{FS}}\big(M, (\{\mathrm{CMT}_i\}_{i=1}^\kappa, \mathbf{A}, \mathbf{u}, R\big) \in \{1,2,3\}^\kappa.$$

4. Let $\Sigma = \Pi_{\mathsf{ring}}$.

$\mathsf{RVerify}_{pp}(M, R, \Sigma)$ : Given $pp = \mathbf{A}$, a message $M$, a ring $R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1})$, and a signature $\Sigma$, this algorithm proceeds as follows:

1. Run algorithm $\mathsf{TAcc}_{\mathbf{A}}(R)$ to compute the root $\mathbf{u}$ of the tree.

2. Parse $\Sigma$ as $\Sigma = (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, (Ch_1, \ldots, Ch_\kappa), \{\mathrm{RSP}\}_{i=1}^{\kappa})$. Return 0 if $(Ch_1, \ldots, Ch_\kappa) \neq \mathcal{H}_{\mathsf{FS}}\big(M, (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathbf{A}, \mathbf{u}, R\big)$.
3. For each $i = 1$ to $\kappa$, run the verification phase of the protocol from Section 4.2 with public input $(\mathbf{A}, \mathbf{u})$ to check the validity of $\mathrm{RSP}_i$ with respect to $\mathrm{CMT}_i$ and $Ch_i$. If any of the conditions does not hold, then return 0. Otherwise, return 1.

## 4.4 Analysis of the Ring Signature Scheme

We first summarize the properties of the given ring signature scheme in the following theorem.

**Theorem 3.** *The ring signature scheme described in Section 4.3 is correct, and produces signatures of bit-size $\widetilde{\mathcal{O}}(n \cdot \log N)$. In the random oracle model, the scheme is unforgeable w.r.t. insider corruption based on the worst-case hardness of the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem, and it is statistically anonymous.*

**Correctness.** The correctness of the ring signature scheme directly follows from the correctness of the accumulator scheme in Section 3 and the perfect completeness of the argument system in Section 4.2: A member of a ring can always obtain a tuple $(\mathbf{x}, \mathbf{d}, w)$ such that $\big((\mathbf{A}, \mathbf{u}), \mathbf{d}, w, \mathbf{x}\big) \in \mathrm{R}_{\mathrm{ring}}$, and thus, his signature on any message always get accepted by the verification algorithm.

**Efficiency.** Since the underlying protocol has communication cost $\widetilde{\mathcal{O}}(\ell \cdot n)$, the signatures produced by the scheme has bit-size $\widetilde{\mathcal{O}}(\kappa \cdot \ell \cdot n) = \widetilde{\mathcal{O}}(\log N \cdot n)$.

**Unforgeability with respect to insider corruption.** For simplicity, the proof of unforgeability assumes that the cardinality of each ring $R^\star$ is a power of 2. However, this restriction can be easily eliminated, as we will see later on.

The proof of unforgeability relies on the following Lemma from [48].

**Lemma 5 ([48],Lemma 8).** *For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a uniformly random $\mathbf{x} \in \{0,1\}^m$, the probability that there exists another $\mathbf{x}' \in \{0,1\}^m \setminus \{\mathbf{x}\}$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{x}' \bmod q$ is at least $1 - 2^{n \cdot \log q - m}$.*

With $m = 2nk$ and $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$, there exists $\mathbf{x}' \in \{0,1\}^m \setminus \{\mathbf{x}\}$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{x}' \bmod q$ with overwhelming probability $1 - 2^{-nk}$.

**Theorem 4.** *The scheme provides unforgeability w.r.t. insider corruption in the random oracle model if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard.* (The proof is available in the full version of the paper.)

**Statistical anonymity.** The proof of the following theorem relies on the statistical witness indistinguishability of the argument system of Lemma 4. The proof is straightforward and omitted.

**Theorem 5.** *The scheme provides statistical anonymity in the random oracle model.*

*Remark 1.* As already mentioned, we can handle arbitrary ring sizes. To this end, one option is to add dummy ring members $\mathbf{d}_{\mathsf{fake},1}, \ldots, \mathbf{d}_{\mathsf{fake},r_0}$ whose public keys are sampled obliviously of their private keys, by deriving them as $\mathbf{d}_{\mathsf{fake},j} = \mathsf{bin}(\mathcal{G}_0(j)) \in \{0,1\}^{nk}$ for each $j \in \{1, \ldots, r_0\}$, where $\mathcal{G}_0 : \mathbb{N} \to \mathbb{Z}_q^n$ is an additional random oracle. A simpler solution is to duplicate one of the actual ring members until reaching a multi-set whose cardinality is a power of two.

## 5 A Lattice-Based Group Signature without Trapdoors

This section shows how to use our accumulator and argument systems to build a lattice-based group signature which is dramatically more efficient than previous proposals as it does not use any trapdoor. Indeed, surprisingly, the scheme does not rely on a standard digital signature to generate group members' private keys.

### 5.1 Definitions

We recall the standard definitions and security requirements for static group signatures [8]. A group signature scheme is a tuple of 4 polynomial-time algorithms $(\mathsf{GKeygen}, \mathsf{GSign}, \mathsf{GVerify}, \mathsf{GOpen})$ defined as follows:

- $\mathsf{GKeygen}$: This is a probabilistic algorithm that takes as input $1^n, 1^N$, where $n \in \mathbb{N}$ is the security parameter and $N \in \mathbb{N}$ is the number of group users, and outputs a triple $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$, where $\mathsf{gpk}$ is the group public key; $\mathsf{gmsk}$ is the group manager's secret key; and $\mathbf{gsk} = (\mathsf{gsk}[0], \ldots, \mathsf{gsk}[N-1])$, where for $j \in \{0, \ldots, N-1\}$, $\mathsf{gsk}[j]$ is the secret key for the group user of index $j$.
- $\mathsf{GSign}$: is a randomized algorithm that inputs $\mathsf{gpk}$, a secret key $\mathsf{gsk}[j]$ for some $j \in \{0, \ldots, N-1\}$, and a message $M$. It returns a group signature $\Sigma$ on $M$.
- $\mathsf{GVerify}$: This deterministic algorithm takes as input the group public key $\mathsf{gpk}$, a message $M$, a purported signature $\Sigma$ on $M$, and returns either 1 or 0.
- $\mathsf{GOpen}$: This deterministic algorithm takes as input the group public key $\mathsf{gpk}$, the group manager's secret key $\mathsf{gmsk}$, a message $M$, a signature $\Sigma$ on $M$, and returns an index $j \in \{0, \ldots, N-1\}$, or $\perp$ (to indicate failure).

*Correctness.* The correctness requirement is stated as follows. For all $n, N \in \mathbb{N}$, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$ produced by $\mathsf{GKeygen}(1^n, 1^N)$, all $j \in \{0, \ldots, N-1\}$, and any message $M \in \{0,1\}^*$, we have $\mathsf{GVerify}\big(\mathsf{gpk}, M, \mathsf{GSign}(\mathsf{gpk}, \mathsf{gsk}[j], M)\big) = 1$ and $\mathsf{GOpen}\big(\mathsf{gpk}, \mathsf{gmsk}, M, \mathsf{GSign}(\mathsf{gsk}[j], M)\big) = j$.

In static groups, the security model of Bellare, Micciancio and Warinschi subsumes the desirable security properties of group signatures using two security notions called *full anonymity* and *full traceability*.

```
$\mathbf{Exp}^{\mathsf{anon}\text{-}\mathsf{b}}_{\mathcal{GS},\mathcal{A}}(n,N)$
$(\mathsf{gpk},\mathsf{gmsk},\mathbf{gsk})$
    $\leftarrow \mathsf{GKeyGen}(1^n,1^N)$
$(\mathsf{st},j_0,j_1,M^\star)$
    $\leftarrow \mathcal{A}_1^{\mathcal{GS}.\mathsf{GOpen}(\mathsf{gpk},\mathsf{msk},.,.)}(\mathsf{gpk},\mathbf{gsk})$
$\Sigma^\star \leftarrow \mathsf{GSign}(\mathsf{gpk},\mathsf{gsk}[j_b],M^\star)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{GS}.\mathsf{GOpen}(\mathsf{gpk},\mathsf{msk},.,.),\neg(M^\star,\Sigma^\star)}(\mathsf{st},\Sigma^\star)$
Return $b'$
```

```
$\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n,N)$
$(\mathsf{gpk},\mathsf{gmsk},\mathbf{gsk}) \leftarrow \mathsf{GKeygen}(1^n,1^N)$
$\mathsf{st} \leftarrow (\mathsf{gmsk},\mathsf{gpk})$
$\mathcal{C} \leftarrow \emptyset$ ; $K \leftarrow \varepsilon$ ; $Cont \leftarrow \mathsf{true}$
while $(Cont = \mathsf{true})$ do
  $(Cont,\mathsf{st},j) \leftarrow$
$\mathcal{A}_1^{\mathcal{GS}.\mathsf{GSign}(\mathsf{gpk},\mathsf{gsk}[\cdot],\cdot)}(\mathsf{st},K)$
    if $Cont = \mathtt{true}$ then $\mathcal{C} \leftarrow \mathcal{C} \cup \{j\}$;
    $K \leftarrow \mathsf{gsk}[j]$
  end if
end while;
$(M^\star,\Sigma^\star) \leftarrow \mathcal{A}_2^{\mathcal{GS}.\mathsf{GSign}(\mathsf{gpk},\mathsf{gsk}[\cdot],\cdot)}(\mathtt{st})$
if $\mathsf{GVerify}(\mathsf{gpk},M^\star,\Sigma^\star) = 0$, Return 0
if $\mathsf{GOpen}(\mathsf{gpk},\mathsf{gmsk},M^\star,\Sigma^\star) = \perp$,
    Return 1
if $\mathsf{GOpen}(\mathsf{gpk},\mathsf{gmsk},M^\star,\Sigma^\star) = j^\star$
    $\wedge\ (j^\star \in \{0,\ldots,N-1\} \setminus \mathcal{C})$
    $\wedge$ (no signing query involved$(j^\star,M^\star)$)
then Return 1 else Return 0
```

**Fig. 3:** Experiments for the definitions of anonymity and full traceability

*Full anonymity.* Full anonymity requires that, without the group manager's secret key, no efficient adversary can infer the identity of a user from its signatures. The adversary should even be unable to distinguish signatures from two distinct users $j_0, j_1$, even knowing their private keys $\mathsf{gsk}[j_0], \mathsf{gsk}[j_1]$. Moreover, this should remain true even when the adversary is granted access to an oracle that opens arbitrary message-signature pairs $(M,\Sigma) \neq (M^\star,\Sigma^\star)$, where $(M^\star,\Sigma^\star)$ is the challenge pair generated by the challenger on behalf of user $j_b$, for some $b \in \{0,1\}$. Formally, the attacker, modeled as a two-stage adversary $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$, is run in the first experiment depicted in Figure 3. The adversary's advantage is defined as

$$\mathbf{Adv}^{\mathsf{anon}}_{\mathcal{GS},\mathcal{A}}(n,N) = \left|\Pr[\mathbf{Exp}^{\mathsf{anon}\text{-}1}_{\mathcal{GS},\mathcal{A}}(n,N)=1] - \Pr[\mathbf{Exp}^{\mathsf{anon}\text{-}0}_{\mathcal{GS},\mathcal{A}}(n,N)=1]\right| .$$

**Definition 10 (Full anonymity, [8]).** *A group signature is* fully anonymous *if, for any polynomial $N$ and any PPT adversary $\mathcal{A}$, $\boldsymbol{Adv}^{\mathsf{anon}}_{\mathcal{GS},\mathcal{A}}(n,N)$ is a negligible function in the security parameter $n$.*

*Full traceability.* Full traceability mandates that all signatures, even those created by colluding users *and* the group manager who pool their secrets together, be traceable to a member of the coalition. The attacker is modeled as a two-stage adversary $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$ which is run in the second experiment of Figure 3, where it is further granted access to an oracle $\mathcal{GS}.\mathsf{GSign}(\mathsf{gpk},\mathsf{gsk}[\cdot],\cdot)$ that returns signatures on behalf of any honest group member. Its success probability against $\mathcal{GS}$ is measured as

$$\mathbf{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n,N) = \Pr[\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n,N) = 1].$$

**Definition 11 (Full traceability, [8]).** *A group signature scheme $\mathcal{GS}$ is* fully traceable *if for any polynomial $N$ and any PPT adversariy $\mathcal{A}$, the probability $\boldsymbol{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n,N)$ is negligible in the security parameter $n$.*

### 5.2 The Underlying Zero-Knowledge Protocol

The group signature scheme that we will present in Section 5.3 relies on an extension of the $\mathsf{ZKAoK}$ in Section 4.2. An encryption layer is added, and the prover additionally has to prove that the given 2 Regev ciphertexts both encrypt the *same* $(j_1,\ldots,j_\ell)^\top$ that was included in $w$. The associated relation is defined as follows.

**Definition 12.** *Define* $\mathrm{R}_{\mathrm{group}} = \Big\{(\mathbf{A},\mathbf{u},\mathbf{B},\mathbf{P}_1,\mathbf{P}_2,\mathbf{c}_1,\mathbf{c}_2),\mathbf{d},w,\mathbf{x},\mathbf{r}_1,\mathbf{r}_2\Big\}$ *as a relation where*

$$\begin{cases}\mathbf{A}\in\mathbb{Z}_q^{n\times m};\ \mathbf{u}\in\{0,1\}^{nk};\ \mathbf{B}\in\mathbb{Z}_p^{n\times m_E};\\ \forall i\in\{1,2\}:\ \mathbf{P}_i\in\mathbb{Z}_p^{\ell\times m_E};\ \mathbf{c}_i=(\mathbf{c}_{i,1},\mathbf{c}_{i,2})\in\mathbb{Z}_p^n\times\mathbb{Z}_p^\ell;\\ \mathbf{d}\in\{0,1\}^{nk};\ w=\big((j_1,\ldots,j_\ell),(\mathbf{w}_\ell,\ldots,\mathbf{w}_1)\big)\in\{0,1\}^\ell\times(\{0,1\}^{nk})^\ell;\\ \mathbf{x}\in\{0,1\}^m;\ \mathbf{r}_1,\mathbf{r}_2\in\{0,1\}^{m_E}\end{cases}$$

*satisfy*

$$\begin{cases}\mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u},\mathbf{d},w\big)=1\ \wedge\ \mathbf{A}\cdot\mathbf{x}=\mathbf{G}\cdot\mathbf{d}\bmod q\\ \forall i\in\{1,2\}:\ \mathbf{c}_{i,1}=\mathbf{B}\cdot\mathbf{r}_i\bmod p\ \wedge\ \mathbf{c}_{i,2}=\mathbf{P}_i\cdot\mathbf{r}_i+\lfloor\frac{p}{2}\rceil\cdot(j_1,\ldots,j_\ell)^\top\bmod p.\end{cases}$$

To prove in $\mathsf{ZK}$ that the vector $(j_1,\ldots,j_\ell)^T$ involved in the new layer is the *same* $(j_1,\ldots,j_\ell)^T$ that was included in $w$, we introduce the following technique.

- For each $c\in\{0,1\}$, let $\mathsf{extbit}(c)=\begin{pmatrix}\bar{c}\\ c\end{pmatrix}\in\{0,1\}^2$.
- For each $b\in\{0,1\}$, we define the permutation $T_b$ that transforms vector $\mathbf{z}=\begin{pmatrix}z_0\\ z_1\end{pmatrix}\in\mathbb{Z}_p^2$ into vector $T_b(\mathbf{z})=\begin{pmatrix}z_b\\ z_{\bar{b}}\end{pmatrix}$.

Observe that the following equivalence holds: For all $b\in\{0,1\}$ and all $\mathbf{z}\in\mathbb{Z}_p^2$,

$$\mathbf{z}=\mathsf{extbit}(j_i)\ \Leftrightarrow\ T_b(\mathbf{z})=\mathsf{extbit}(j_i\oplus b). \tag{14}$$

In Stern's framework, this equivalence allows us to prove in $\mathsf{ZK}$ the possession of the bit $j_i$, for every $i\in[\ell]$, by extending $j_i$ to $\mathsf{extbit}(j_i)$ and then, by permuting it with a one-time pad $b_i$. Furthermore, to prove that the same $j_i$ is involved in both layers, we will use the same one-time pad in both layers of the protocol.

Embedding this new technique into the protocol in Section 4.2, we obtain an argument system for the relation $\mathrm{R}_{\mathrm{group}}$. As for the previous two protocols, they also rely on the string commitment scheme from [40], which is statistically hiding and computationally binding if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard.

**Lemma 6.** *Assume that the* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *problem is hard. Then, there exists a statistical* $\mathsf{ZKAoK}$ *for the relation* $\mathrm{R}_{\mathrm{group}}$ *with perfect completeness and communication cost* $\widetilde{\mathcal{O}}(\ell \cdot n) + \mathcal{O}((m_E + \ell) \cdot \log p)$. *In particular:*

– *There exists an efficient simulator that, on input* $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{c}_1, \mathbf{c}_2)$, *outputs an accepted transcript which is statistically close to that produced by the real prover.*
– *There exists an efficient knowledge extractor that, on input* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to the same commitment* CMT, *outputs* $(\mathbf{d}', w', \mathbf{x}', \mathbf{r}'_1, \mathbf{r}'_2)$ *such that*

$$\big((\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{c}_1, \mathbf{c}_2), \mathbf{d}', w', \mathbf{x}', \mathbf{r}'_1, \mathbf{r}'_2\big) \in \mathrm{R}_{\mathrm{group}}.$$

The full description and analysis of the argument system are given in the full version of the paper.

## 5.3 Our Construction

Let $n$ be the security parameter, and $N = 2^\ell = \mathsf{poly}(n)$ be the maximum expected number of group users. Parameters $m, q, k, \kappa$ and the random oracle $\mathcal{H}_{\mathsf{FS}}$ are defined as in the ring signature scheme in Section 4.3. To employ the $\ell$-bit version of Regev's encryption scheme, we will also need prime modulus $p = \widetilde{\mathcal{O}}(n^{1.5})$, parameter $m_E = 2(n+\ell)\lceil \log p \rceil$, and an $\mathsf{LWE}$ error distribution $\chi = D_{\mathbb{Z}, 2\sqrt{n}}$.

$\mathsf{\textbf{GKeygen}}(1^n, 1^N)$ : This algorithm begins by sampling a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. Then, it performs the following steps:

1. For each $j \in [0, N-1]$, sample a random binary vector $\mathbf{x}_j \xleftarrow{\$} \{0,1\}^m$ and compute $\mathbf{d}_j = \mathsf{bin}(\mathbf{A} \cdot \mathbf{x}_j \bmod q) \in \{0,1\}^{nk}$. In the unlikely event that $\{\mathbf{d}_j\}_{j=0}^{N-1}$ are not pairwise distinct, restart the process. Otherwise, define the set $R = (\mathbf{d}_0, \ldots, \mathbf{d}_{N-1})$.
2. Run algorithm $\mathsf{TAcc}_{\mathbf{A}}(R)$ to build the Merkle tree based on $R$ and the hash function $h_{\mathbf{A}}$, and obtain the root $\mathbf{u} \in \{0,1\}^{nk}$.
3. For each $j \in [0, N-1]$, run algorithm $\mathsf{TWitness}_{\mathbf{A}}(R, \mathbf{d}_j)$ to output a witness
$$w^{(j)} = \big((j_1, \ldots, j_\ell) \in \{0,1\}^\ell, (\mathbf{w}_\ell^{(j)}, \ldots, \mathbf{w}_1^{(j)}) \in (\{0,1\}^{nk})^\ell\big)$$
to the fact that $\mathbf{d}_j$ was accumulated in $\mathbf{u}$. (Note that $(j_1, \ldots, j_\ell)$ is the binary representation of $j$.) Then define $\mathsf{gsk}[j] = (\mathbf{x}_j, \mathbf{d}_j, w^{(j)})$.
4. Sample $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_p^{n \times m_E}$. For $i \in \{1,2\}$, sample $\mathbf{S}_i \xleftarrow{\$} \mathbb{Z}_p^{n \times \ell}$, $\mathbf{E}_i \hookleftarrow \chi^{\ell \times m_E}$, and compute $\mathbf{P}_i = \mathbf{S}_i^\top \cdot \mathbf{B} + \mathbf{E}_i \in \mathbb{Z}_p^{\ell \times m_E}$.
5. Output
$$\mathsf{gpk} := \{\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2\}\,; \;\; \mathsf{gmsk} := \mathbf{S}_1; \;\; \mathbf{gsk} := (\mathsf{gsk}[0], \ldots, \mathsf{gsk}[N-1]).$$

$\mathsf{\textbf{GSign}}(\mathsf{gpk}, \mathsf{gsk}[j], M)$: To sign $M \in \{0,1\}^*$ using $\mathsf{gsk}[j] = (\mathbf{x}_j, \mathbf{d}_j, w^{(j)})$, where $w^{(j)} = \big((j_1, \ldots, j_\ell), (\mathbf{w}_\ell^{(j)}, \ldots, \mathbf{w}_1^{(j)})\big)$, the user conducts the following steps:

1. Encrypt $(j_1, \ldots, j_\ell) \in \{0,1\}^\ell$ twice using Regev's encryption scheme. Namely, for each $i \in \{1,2\}$, sample $\mathbf{r}_i \xleftarrow{\$} \{0,1\}^{m_E}$ and compute

$$\begin{aligned}\mathbf{c}_i &= (\mathbf{c}_{i,1}, \mathbf{c}_{i,2}) \\ &= \Big(\mathbf{B} \cdot \mathbf{r}_i \bmod p,\ \mathbf{P}_i \cdot \mathbf{r}_i + \lceil \frac{p}{2} \rfloor \cdot (j_1, \ldots, j_\ell)^\top \bmod p\Big) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^\ell.\end{aligned}$$

2. Generate a $\mathsf{NIZKAoK}$ $\Pi_{\mathsf{group}}$ in order to demonstrate the possession of a valid tuple $\tau = \big(\mathbf{x}_j, \mathbf{d}_j, w^{(j)}, \mathbf{r}_1, \mathbf{r}_2\big)$, where $w^{(j)} = \big((j_1, \ldots, j_\ell), (\mathbf{w}_\ell^{(j)}, \ldots, \mathbf{w}_1^{(j)})\big)$, such that:
   (a) $\mathbf{A} \cdot \mathbf{x}_j = \mathbf{G} \cdot \mathbf{d}_j \bmod q$ and $\mathsf{TVerify}_{\mathbf{A}}\big(\mathbf{u}, \mathbf{d}_j, w^{(j)}\big) = 1$.
   (b) $\mathbf{c}_1$ and $\mathbf{c}_2$ are both correct encryptions of $(j_1, \ldots, j_\ell)$ with randomness $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively.

   This is done by running the protocol in Section 5.2 with public input $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{c}_1, \mathbf{c}_2)$ and prover's witness $\tau$ defined above. The protocol is repeated $\kappa = \omega(\log n)$ times to achieve negligible soundness error and made non-interactive via the Fiat-Shamir heuristic as a triple $\Pi_{\mathsf{group}} = (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathrm{CH}, \{\mathrm{RSP}\}_{i=1}^{\kappa})$, where

$$\mathrm{CH} = \mathcal{H}_{\mathsf{FS}}\big(M, (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{c}_1, \mathbf{c}_2\big) \in \{1,2,3\}^{\kappa}.$$

3. Output the group signature $\Sigma = (\Pi_{\mathsf{group}}, \mathbf{c}_1, \mathbf{c}_2)$.

**$\mathsf{GVerify}$**$(\mathsf{gpk}, M, \Sigma)$ : This algorithm proceeds as follows:

1. Parse $\Sigma$ as $\Sigma = \big(\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, (Ch_1, \ldots, Ch_\kappa), \{\mathrm{RSP}\}_{i=1}^{\kappa}, \mathbf{c}_1, \mathbf{c}_2\big)$.
   If $(Ch_1, \ldots, Ch_\kappa) \neq \mathcal{H}_{\mathsf{FS}}\big(M, (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{c}_1, \mathbf{c}_2\big)$, then return 0.
2. For each $i = 1$ to $\kappa$, run the verification phase of the protocol in Section 5.2 with public input $(\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{c}_1, \mathbf{c}_2)$ to check the validity of $\mathrm{RSP}_i$ w.r.t. $\mathrm{CMT}_i$ and $Ch_i$. If any of the conditions does not hold, then return 0.
3. Return 1.

**$\mathsf{GOpen}$**$(\mathsf{gpk}, \mathsf{gmsk}, \Sigma, M)$**:** On input $\mathsf{gmsk} = \mathbf{S}_1$ and a group signature $\Sigma = (\Pi_{\mathsf{group}}, \mathbf{c}_1, \mathbf{c}_2)$ on message $M$, this algorithm decrypts $\mathbf{c}_1 = (\mathbf{c}_{1,1}, \mathbf{c}_{1,2})$ and returns an index $j \in [0, N-1]$, as follows:

1. Compute $(j_1', \ldots, j_\ell') = \mathbf{c}_{1,2} - \mathbf{S}_1^\top \cdot \mathbf{c}_{1,1} \in \mathbb{Z}_p^\ell$.
2. For each $i \in [\ell]$, if $j_i'$ is closer to 0 than to $\lceil \frac{p}{2} \rfloor$ modulo $p$, then let $j_i = 0$; otherwise, let $j_i = 1$.
3. Output index $j \in [0, N-1]$ that has binary representation $(j_1, \ldots, j_\ell)$.

**Efficiency.** The public key consists of a constant number of matrices over $\mathbb{Z}_q$ and $\mathbb{Z}_p$, where $q$ and $p$ are small moduli. The group signature has bit-size $\kappa \cdot \big(\widetilde{\mathcal{O}}(\ell \cdot n) + \mathcal{O}((m_E + \ell) \cdot \log p)\big) = \widetilde{\mathcal{O}}(\log N \cdot n)$. The scheme is dramatically more efficient than previous lattice-based realizations of group signatures. Indeed, its most important advantage is that it does not require any party to hold a GPV trapdoor. As observed by Lyubashevsky [49], lattice-based signatures without trapdoor can be made significantly more efficient.

**Correctness.** The correctness of algorithm $\mathsf{GVerify}$ follows directly from the correctness of the accumulator scheme in Section 3, and the completeness of the argument system in Section 5.2. As for the correctness of algorithm $\mathsf{GOpen}$, it suffices to note that

$$\begin{aligned}\mathbf{c}_{1,2} - \mathbf{S}_1^\top \cdot \mathbf{c}_{1,1} &= (\mathbf{S}_1^\top \cdot \mathbf{B} + \mathbf{E}_1) \cdot \mathbf{r}_1 + \lceil \frac{p}{2} \rfloor \cdot (j_1, \ldots, j_\ell)^\top - \mathbf{S}_1^\top \cdot \mathbf{B} \cdot \mathbf{r}_1 \\ &= \mathbf{E}_1 \cdot \mathbf{r}_1 + \lceil \frac{p}{2} \rfloor \cdot (j_1, \ldots, j_\ell)^\top \bmod p,\end{aligned}$$

and $\|\mathbf{E}_1 \cdot \mathbf{r}_1\|_\infty < p/4$ with overwhelming probability, for the given setting of parameters, and the decryption algorithm should return $(j_1, \ldots, j_\ell)^\top$.

**Security.** The full traceability property of our scheme is stated in Theorem 6. In the proof, which is given in the full version of the paper we prove that any adversary with noticeable probability of evading traceability implies an algorithm for either breaking the security of the underlying accumulator of Section 3, breaking the computational soundness of the argument system in Section 5.2, or solving an instance of the $\mathsf{SIS}^\infty_{n,m,q,1}$ problem.

**Theorem 6.** *The scheme provides full traceability in the random oracle model if the* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *problem is hard.*

The proof of full anonymity relies on the fact that applying the Naor-Yung paradigm [56] to Regev's cryptosystem yields an IND-CCA2 secure cryptosystem. (A similar argument was used by Benhamouda *et al.* [12] for an NTRU-like encryption scheme.) Indeed, the argument system of Definition 12 implies that $\mathbf{c}_1$ and $\mathbf{c}_2$ encrypt the same message. In the random oracle model, it was already observed by Fouque and Pointcheval [30] (see [13] for a more general treatment) that applying the Fiat-Shamir heuristic to $\Sigma$-protocols gives simulation-sound proofs [66]. Similarly to [30,13], the proof of Theorem 7 relies on the fact that applying Fiat-Shamir to the argument system of Definition 12 yields a simulation-sound NIZK argument in the random oracle model if the underlying commitment is computationally binding. This holds even though this argument system does not have the standard special soundness property (*i.e.*, three accepting conversations for distinct challenges are necessary to extract a witness). Simulation-soundness is actually implied by Lemma 6: suppose that $\mathbf{c}_1$ and $\mathbf{c}_2$ encrypt distinct $\ell$-bit strings. This means that there exists no vector $(\mathbf{r}_1^T \mid \mathbf{r}_2^T)^T$ such that

$$\left[\begin{array}{c|c} \mathbf{B} & -\mathbf{B} \\ \hline \mathbf{P}_1 & -\mathbf{P}_2 \end{array}\right] \cdot \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{1,1} - \mathbf{c}_{2,1} \\ \mathbf{c}_{2,1} - \mathbf{c}_{2,2} \end{bmatrix}.$$

Now, recall that the computational soundness of all Stern-type protocols is proved by showing that the knowledge extractor obtains either a set of valid witnesses or breaks the binding property of the underlying commitment. Given that the witnesses do not exist if the statement is false, by rewinding a simulation-soundness adversary sufficiently many times, the knowledge extractor necessarily extracts two openings of a given commitment.

The proof of Theorem 7 is similar to [66] and given in the full version of the paper.

**Theorem 7.** *The scheme provides full anonymity if the* $\mathsf{LWE}_{n,p,\chi}$ *problem is hard, and if the argument system is simulation-sound.*

## Acknowledgements

We thank Damien Stehlé for useful discussions and the anonymous reviewers of EUROCRYPT 2016 for helpful 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".

## References

1. T. Acar and L. Nguyen. Revocation for Delegatable Anonymous Credentials. In *PKC 2011*, volume 6571 of *LNCS*, pages 423–440. Springer, 2011.
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. M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *STOC 1996*, pages 99–108. ACM, 1996.
4. M. Ajtai. Generating Hard Instances of the Short Basis Problem. In *ICALP 1999*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
5. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
6. M. H. Au, Q. Wu, W. Susilo, and Y. Mu. Compact E-Cash from Bounded Accumulator. In *CT-RSA 2007*, volume 4377 of *LNCS*, pages 178–195. Springer, 2007.
7. N. Baric and B. Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494. Springer, 1997.
8. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.
9. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. In *IEEE S&P 2014*, pages 459–474. IEEE, 2014.
10. J. Benaloh and M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures. In *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285. Springer, 1993.

11. A. Bender, J. Katz, and R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *J. Cryptology*, 22(1):114–138, 2009.
12. 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*, volume 8873 of *LNCS*, pages 551–572. Springer, 2014.
13. D. Bernhard, M. Fischlin, and B. Warinschi. Adaptive Proofs of Knowledge in the Random Oracle Model. In *PKC 2015*, volume 9020 of *LNCS*, pages 629–649. Springer, 2015.
14. D. Boneh and X. Boyen. Short Signatures Without Random Oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
15. D. Boneh and H. Corrigan-Gibbs. Bivariate Polynomials Modulo Composites and Their Applications. In *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 42–62. Springer, 2014.
16. J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short Accountable Ring Signatures Based on DDH. In *ESORICS 2015*, volume 9326 of *LNCS*. Springer, 2015.
17. 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.
18. Z. Brakerski and Y. T. Kalai. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. *IACR Cryptology ePrint Archive*, 2010:86, 2010.
19. J. Camenisch, M. Kohlweiss, and C. Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC 2009*, volume 5443 of *LNCS*, pages 481–500. Springer, 2009.
20. J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
21. J. Camenisch, G. Neven, and M. Rückert. Fully Anonymous Attribute Tokens from Lattices. In *SCN 2012*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012.
22. S. Canard and A. Gouget. Multiple Denominations in E-cash with Compact Transaction Data. In *FC 2010*, volume 6052 of *LNCS*, pages 82–97. Springer, 2010.
23. 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.
24. D. Catalano and D. Fiore. Vector Commitments and Their Applications. In *PKC 2013*, volume 7778 of *LNCS*, pages 55–72. Springer, 2013.
25. N. Chandran, J. Groth, and A. Sahai. Ring Signatures of Sub-linear Size Without Random Oracles. In *ICALP 2007*, volume 4596 of *LNCS*, pages 423–434. Springer, 2007.
26. D. Chaum and E. van Heyst. Group Signatures. In *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
27. D. Derler, C. Hanser, and D. Slamanig. Revisiting Cryptographic Accumulators, Additional Properties and Relations to Other Primitives. In *CT-RSA 2015*, volume 9048 of *LNCS*, pages 127–144. Springer, 2015.
28. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous Identification in Ad Hoc Groups. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 609–626. Springer, 2004.

29. 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, Part I*, volume 9452 of *LNCS*, pages 260–285. Springer, 2015.
30. P.-A. Fouque and D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 351–368. Springer, 2001.
31. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC 2008*, pages 197–206. ACM, 2008.
32. O. Goldreich, S. Goldwasser, and S. Halevi. Collision-Free Hashing from Lattice Problems. *ECCC*, 3(42), 1996.
33. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A Group Signature Scheme from Lattice Assumptions. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 395–412. Springer, 2010.
34. J. Groth. Evaluating Security of Voting Schemes in the Universal Composability Framework. In *ACNS 2004*, volume 3089 of *LNCS*, pages 46–60. Springer, 2004.
35. J. Groth. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, 2010.
36. J. Groth and M. Kohlweiss. One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin. In *EUROCRYPT 2015*, volume 9057 of *LNCS*, pages 253–280. Springer, 2015.
37. 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.
38. M. P. Jhanwar and R. Safavi-Naini. Compact Accumulator using Lattices. IACR Cryptology ePrint Archive: Report 2014/1015, February 2015.
39. A. Kawachi, K. Tanaka, and K. Xagawa. Multi-bit Cryptosystems Based on Lattice Problems. In *PKC 2007*, volume 4450 of *LNCS*, pages 315–329. Springer, 2007.
40. 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.
41. 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.
42. 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.
43. J. Li, N. Li, and R. Xue. Universal Accumulators with Efficient Nonmembership Proofs. In *ACNS 2007*, volume 4521 of *LNCS*, pages 253–269. Springer, 2007.
44. Z. Lin and N. Hopper. Jack: Scalable Accumulator-Based Nymble System. In *WPES 2010*, pages 53–62. ACM, 2010.
45. 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 of *LNCS*, pages 107–124. Springer, 2013.
46. 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.
47. H. Lipmaa. Secure Accumulators from Euclidean Rings Without Trusted Setup. In *ACNS 2012*, volume 7341 of *LNCS*, pages 224–240. Springer, 2012.
48. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.

49. V. Lyubashevsky. Lattice Signatures without Trapdoors. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.
50. R. C. Merkle. A Certified Digital Signature. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 218–238. Springer, 1989.
51. D. Micciancio and P. Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, 2011.
52. D. Micciancio and C. Peikert. Hardness of SIS and LWE with Small Parameters. In *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 21–39. Springer, 2013.
53. D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.
54. I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *IEEE S&P 2013*, pages 397–411. IEEE, 2013.
55. M. Naor. On Cryptographic Assumptions and Challenges. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, 2003.
56. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC 1990*, pages 427–437. ACM, 1990.
57. L. Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer, 2005.
58. 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.
59. C. Papamanthou, E. Shi, R. Tamassia, and K. Yi. Streaming Authenticated Data Structures. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 353–370. Springer, 2013.
60. C. Papamanthou, R. Tamassia, and N. Triandopoulos. Authenticated Hash Tables. In *ACM-CCS 2008*, pages 437–448. ACM, 2008.
61. C. Peikert. Public-key Cryptosystems from the Worst-case Shortest Vector Problem: Extended Abstract. In *STOC 2009*, pages 333–342. ACM, 2009.
62. C. Peikert, V. Vaikuntanathan, and B. Waters. A Framework for Efficient and Composable Oblivious Transfer. In *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008.
63. M. Prabhakaran and R. Xue. Statistically Hiding Sets. In *CT-RSA 2009*, volume 5473 of *LNCS*, pages 100–116. Springer, 2009.
64. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.
65. R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.
66. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS 1999*, pages 543–553, 1999.
67. J. Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.
68. G. Tsudik and S. Xu. Accumulating Composites and Improved Group Signing. In *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 269–286. Springer, 2003.
69. R. Xue, N. Li, and J. Li. Algebraic Construction for Zero-Knowledge Sets. *J. Comput. Sci. Technol.*, 23(2):166–175, 2008.