Lattice-Based Group Signatures with Logarithmic Signature Size

Fabien Laguillaumie[1,3], Adeline Langlois[2,3], Benoît Libert[4], and Damien Stehlé[2,3]

[1] Université Claude Bernard Lyon 1
[2] École Normale Supérieure de Lyon
[3] LIP (U. Lyon, CNRS, ENS Lyon, INRIA, UCBL), 46 Allée d'Italie, 69364 Lyon Cedex 07, France.
[4] Technicolor, 975 Avenue des Champs Blancs, 35510 Cesson-Sévigné, France

Abstract. Group signatures are cryptographic primitives where users can anonymously sign messages in the name of a population they belong to. Gordon *et al.* (Asiacrypt 2010) suggested the first realization of group signatures based on lattice assumptions in the random oracle model. A significant drawback of their scheme is its linear signature size in the cardinality N of the group. A recent extension proposed by Camenisch *et al.* (SCN 2012) suffers from the same overhead. In this paper, we describe the first lattice-based group signature schemes where the signature and public key sizes are essentially logarithmic in N (for any fixed security level). Our basic construction only satisfies a relaxed definition of anonymity (just like the Gordon *et al.* system) but readily extends into a fully anonymous group signature (*i.e.*, that resists adversaries equipped with a signature opening oracle). We prove the security of our schemes in the random oracle model under the SIS and LWE assumptions.

Keywords. Lattice-based cryptography, group signatures, anonymity.

1 Introduction

Group signatures are a core cryptographic primitive that paradoxically combines the properties of authenticity and anonymity. They are useful in many real-life applications including trusted computing platforms, auction protocols or privacy-protecting mechanisms for users in public transportation.

Parties involved in such a system are a special entity, called the group manager, and group members. The manager holds a master secret key, generates a system-wide public key, and administers the group members, by providing to each of them an individual secret key that will allow them to anonymously sign on behalf of the group. In case of dispute, the manager (or a separate authority) is able to determine the identity of a signer via an opening operation. This fundamental primitive has been extensively studied, from both theoretical and practical perspectives: It has been enriched with many useful properties, and it has been implemented in the contexts of trusted computing (using privacy-preserving attestation [12]) and of traffic management (e.g., the Vehicle Safety Communications project of the U.S. Dept. of Transportation [29]).

Group signatures were originally proposed by Chaum and van Heyst [18] and made scalable by Ateniese *et al.* in [3]. Proper security models were introduced in [5] and [6, 30] (for dynamic groups), whereas more intricate and redundant properties were considered hitherto. The model of Bellare *et al.* [5] requires two main security properties called *full anonymity* and *full traceability.* The former notion means that signatures do not leak the identities of their originators, whereas the latter implies that no collusion of malicious users can produce a valid signature that cannot be traced to one of them. Bellare *et al.* [5] proved that trapdoor permutations suffice to design group signatures, but their theoretical construction was mostly a proof of concept. Nevertheless, their methodology has been adapted in practical constructions: Essentially, a group member signs a message by verifiably encrypting a valid membership certificate delivered by the authority, while hiding its identity. While numerous schemes (e.g., [3, 13, 15, 7]) rely on the random oracle model (ROM), others are proved secure in the standard model (e.g., [5, 6, 9, 10, 25]). Except theoretical constructions [5, 6], all of these rely on the Groth-Sahai methodology to design non-interactive proof systems for specific languages involving elements in bilinear groups [27]. This powerful tool led to the design of elegant compact group signatures [10, 25] whose security relies on pairing-related assumptions. The resulting signatures typically consist in a constant number of elements of a group admitting a secure and efficient bilinear map.

Lattices and Group Signatures. Lattices are emerging as a promising alternative to traditional number-theoretic tools like bilinear maps. They lead to asymptotically faster solutions, thanks to the algorithmic simplicity of the involved operations and to the high cost of the best known attacks. Moreover, lattice-based schemes often enjoy strong security guarantees, thanks to worst-case/average-case connections between lattice problems, and to the conjectured resistance to quantum computers.

While numerous works have been (successfully) harnessing the power of lattices for constructing digital signatures (see [36, 23, 17, 33, 8, 34] and references therein), only two works addressed the problem of efficiently realizing lattice-based group signatures. The main difficulty to overcome is arguably the scarcity of efficient and expressive non-interactive proof systems for statements involving lattices, in particular for statements on the witnesses of the hard average-case lattice problems. This state of affairs contrasts with the situation in bilinear groups, where powerful non-interactive proof systems are available [26, 27].

In 2010, Gordon *et al.* [24] described the first group signature based on lattice assumptions using the Gentry *et al.* signature scheme [23] as membership certificate, an adaptation of Regev's encryption scheme [43] to encrypt it, and a zero-knowledge proof technique due to Micciancio and Vadhan [39]. While elegant in its design principle, their scheme suffers from signatures and public keys of sizes linear in the number of group members, making it utterly inefficient in comparison with constructions based on bilinear maps [7] or the strong RSA assumption [3]. Quite recently, Camenisch *et al.* [16] proposed anonymous attribute token systems, which can be seen as generalizations of group signatures.

One of their schemes improves upon [24] in that the group public key has constant size[5] and the anonymity property is achieved in a stronger model where the adversary is granted access to a signature opening oracle. Unfortunately, all the constructions of [16] inherit the linear signature size of the Gordon *et al.* construction. Thus far, it remained an open problem to break the linear-size barrier. This is an important challenge considering that, as advocated by Bellare *et al.* [5], one should expect practical group signatures not to entail more than poly-logarithmic complexities in the group sizes.

Our Contributions. We describe the first lattice-based group signatures featuring sub-linear signature sizes. If t and N denote the security parameter and the maximal group size, the public keys and signatures are $\widetilde{\mathcal{O}}(t^2 \cdot \log N)$ bit long. Notice that no group signature scheme can provide signatures containing $o(\log N)$ bits (such signatures would be impossible to open), so that the main improvement potential lies in the $\widetilde{\mathcal{O}}(t^2)$ factor. These first asymptotically efficient (in t and $\log N$) lattice-based group signatures are a first step towards a practical alternative to the pairing-based counterparts. The security proofs hold in the ROM (as for [24, 16]), under the Learning With Error (LWE) and Short Integer Solution (SIS) assumptions. While our basic system only provides anonymity in a relaxed model (like [24]) where the adversary has no signature opening oracle, we show how to upgrade it into a fully anonymous group signature, in the anonymity model of Bellare *et al.* [5]. This is achieved at a minimal cost in that the signature length is only increased by a constant factor. In contrast, Camenisch *et al.* [16, Se. 5.2] achieve full anonymity at the expense of inflating their basic signatures by a factor proportional to the security parameter.

Construction Overview. Our construction is inspired by the general paradigm from [5] consisting in *encrypting* a membership certificate under the authority's public key while providing a *non-interactive proof* that the ciphertext encrypts a valid *certificate* belonging to some group member. Nevertheless, our scheme differs from this paradigm in the sense that it is not the certificate itself which is encrypted. Instead, a temporary certificate, produced at each signature generation, is derived from the initial one and encrypted, with a proof of its validity.

We also depart from the approach of [24] at the very core of the design, *i.e.*, when it comes to provide evidence that the encrypted certificate corresponds to a legitimate group member. Specifically, Gordon *et al.* [24] hide their certificate, which is a GPV signature [23, Se. 6], within a set of $N-1$ (encrypted) GPV pseudo-signatures that satisfy the same verification equation without being short vectors. Here, to avoid the $\mathcal{O}(N)$ factor in the signature size, we take a different approach which is reminiscent of the Boyen-Waters group signature [9]. Each group member is assigned a unique ℓ-bit identifier $\mathrm{id} = \mathrm{id}[1] \ldots \mathrm{id}[\ell] \in \{0,1\}^\ell$, where $\ell = \lceil \log_2 N \rceil$. Its certificate is an extension of a Boyen signature [8] consisting in a *full* short basis of a certain lattice (instead of a single vector), which allows the signer to generate *temporary certificates* composed of a pair $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{Z}^m$

[5] This can also be achieved with [24] by replacing the public key by a hash thereof, and appending the key to the signature.

of discrete Gaussian vectors such that

$$\mathbf{x}_1^T \cdot \mathbf{A} + \mathbf{x}_2^T \cdot (\mathbf{A}_0 + \sum_{1 \leq i \leq \ell} \mathrm{id}[i] \cdot \mathbf{A}_i) = \mathbf{0} \bmod q. \tag{1}$$

Here, q is a small bit length integer and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{m \times n}$ are part of the group public key. Our choice of Boyen's signature [8] as membership certificate is justified by it being one of the most efficient known lattice-based signatures proven secure in the standard model, and enjoying a simple verification procedure corresponding to a relation for which we can design a proof of knowledge. A signature proven secure in the standard model allows us to obtain an easy-to-prove relation that does not involve a random oracle. As noted for example in [3, 14, 15], signature schemes outside the ROM make it easier to prove knowledge of a valid message-signature pair in the design of privacy-preserving protocols.

We encrypt $\mathbf{x}_2 \in \mathbb{Z}^m$ as in [24], using a variant of the dual-Regev encryption scheme [23, Se. 7]: the resulting ciphertext is $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s} + \mathbf{x}_2$, where $\mathbf{B}_0 \in \mathbb{Z}_q^{m \times n}$ is a public matrix and $\mathbf{s}$ is uniform in $\mathbb{Z}_q^n$. Then, for each $i \in [1, \ell]$, we also compute a proper dual-Regev encryption $\mathbf{c}_i$ of $\mathrm{id}[i] \cdot \mathbf{x}_2$ and generate a non-interactive OR proof that $\mathbf{c}_i$ encrypts either the same vector as $\mathbf{c}_0$ or the $\mathbf{0}$ vector.

It remains to prove that the encrypted vectors $\mathbf{x}_2$ are part of a signature satisfying (1) without giving away the $\mathrm{id}[i]$'s. To this end, we choose the signing matrices $\mathbf{A}_i$ orthogonally to the encrypting matrices $\mathbf{B}_i$, as suggested in [24]. Contrarily to the case of [24], the latter technique does not by itself suffice to guarantee the well-formedness of the $\mathbf{c}_i$'s. Indeed, we also need to prove properties about the noise vectors used in the dual-Regev ciphertexts $\{\mathbf{c}_i\}_{1 \leq i \leq \ell}$. This is achieved using a modification of Lyubashevsky's protocol [32, 34] to prove knowledge of a solution to the Inhomogeneous Short Integer Solution problem (ISIS). This modification leads to a Σ-protocol which is zero-knowledge when the transcript is conditioned on the protocol not aborting. As the challenge space of this Σ-protocol is binary, we lowered the abort probability so that we can efficiently apply the Fiat-Shamir heuristic to a parallel repetition of the basic protocol. In the traceability proof, the existence of a witness extractor will guarantee that a successful forger will either yield a forgery for Boyen's signature or a short non-zero vector in the kernel of one of the matrices $\{\mathbf{A}_i\}_{1 \leq i \leq \ell}$. In either case, the forger allows the simulator to solve a SIS instance.

In the fully anonymous variant of our scheme, the difficulty is to find a way to open adversarially-chosen signatures. This is achieved by implicitly using a "chosen-ciphertext-secure" variant of the signature encryption technique of Gordon *et al.* [24]. While Camenisch *et al.* [16] proceed in a similar way using Peikert's technique [40], we use a much more economical method borrowed from the Agrawal *et al.* [1] identity-based cryptosystem. In our basic system, each $\mathbf{c}_i$ is of the form $\mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}[i] \cdot \mathbf{x}_2$, where p is an upper bound on $\mathbf{x}_2$'s coordinates, and can be decrypted using a short basis $\mathbf{S}_i$ such that $\mathbf{S}_i \cdot \mathbf{B}_i = \mathbf{0} \bmod q$. Our fully anonymous system replaces each $\mathbf{B}_i$ by a matrix $\mathbf{B}_{i,\mathsf{VK}}$ that depends on the verification key VK of a one-time signature. In the proof of full anonymity, the reduction will be able to compute a trapdoor for all matrices $\mathbf{B}_{i,\mathsf{VK}}$, except for one specific verification key $\mathsf{VK}^\star$ that will be used in the challenge phase. This will provide the reduction with a backdoor allowing it to open all adversarially-generated signatures.

Open problems. The schemes we proposed should be viewed as proofs of concept, since instantiating them with practical parameters would most likely lead to large keys and signature sizes. It is an interesting task to replace the SIS and LWE problems by their ring variants [35, 41, 37], to attempt to save linear factors in the security parameter t. The main hurdle in that direction seems to be the design of appropriate zero-knowledge proofs of knowledge for the LWE and ISIS relations (see Section 2.2).

As opposed to many pairing-based constructions, the security of our scheme is only proven in the random oracle model: We rely on the Fiat-Shamir heuristic to remove the interaction in the interactive proof systems. This is because very few lattice problems are known to belong to NIZK. The problems considered in the sole work on this topic [42] seem ill-fitted to devise group signatures. As a consequence, the security proofs of all known lattice-based group signatures are conducted in the random oracle model. Recently suggested multi-linear maps [22] seem like a possible direction towards solving this problem. However, currently known instantiations [22, 19] rely on assumptions that seem stronger than LWE or SIS.

2 Background and Definitions

We first recall standard notations. All vectors will be denoted in bold lower-case letters, whereas bold upper-case letters will be used for matrices. If $\mathbf{b}$ and $\mathbf{c}$ are two vectors of compatible dimensions and base rings, then their inner product will be denoted by $\langle \mathbf{b}, \mathbf{c} \rangle$. Further, if $\mathbf{b} \in \mathbb{R}^n$, its euclidean norm will be denoted by $\|\mathbf{b}\|$. This notation is extended to any matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ with columns $(\mathbf{b}_i)_{i \leq n}$ by $\|\mathbf{B}\| = \max_{i \leq n} \|\mathbf{b}_i\|$. If $\mathbf{B}$ is full column-rank, we let $\widetilde{\mathbf{B}}$ denote the Gram-Schmidt orthogonalisation of $\mathbf{B}$.

If D_1 and D_2 are two distributions over the same countable support S, then their statistical distance is defined as $\Delta(D_1, D_2) = \frac{1}{2} \sum_{x \in S} |D_1(x) - D_2(x)|$. A function $f(n)$ is said negligible if $f(n) = n^{-\omega(1)}$. Finally, the acronym PPT stands for probabilistic polynomial-time.

2.1 Lattices

A (full-rank) lattice L is the set of all integer linear combinations of some linearly independent basis vectors $(\mathbf{b}_i)_{i \leq n}$ belonging to some $\mathbb{R}^n$. For a lattice L and a real $\sigma > 0$, we define the Gaussian distribution of support L and parameter σ by $D_{L,\sigma}[\mathbf{b}] \sim \exp(-\pi \|\mathbf{b}\|^2 / \sigma^2)$, for all $\mathbf{b}$ in L. We will extensively use the fact that samples from $D_{L,\sigma}$ are short with overwhelming probability.

Lemma 1 ([4, Le. 1.5]). *For any lattice $L \subseteq \mathbb{R}^n$ and $\sigma > 0$, we have $\Pr_{\mathbf{b} \hookleftarrow D_{L,\sigma}}[\|\mathbf{b}\| \leq \sqrt{n}\sigma] \geq 1 - 2^{-\Omega(n)}$.*

As shown by Gentry *et al.* [23], Gaussian distributions with lattice support can be sampled from efficiently, given a sufficiently short basis of the lattice.

Lemma 2 ([11, Le. 2.3]). *There exists a* PPT *algorithm* GPVSample *that takes as inputs a basis* $\mathbf{B}$ *of a lattice* $L \subseteq \mathbb{Z}^n$ *and a rational* $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$*, and outputs vectors* $\mathbf{b} \in L$ *with distribution* $D_{L,\sigma}$*.*

Cash *et al.* [17] showed how to use GPVSample to randomize the basis of a given lattice. The following statement is obtained by using [11, Le. 2.3] in the proof of [17].

Lemma 3 (Adapted from [17, Le. 3.3]). *There exists a* PPT *algorithm* RandBasis *that takes as inputs a basis* $\mathbf{B}$ *of a lattice* $L \subseteq \mathbb{Z}^n$ *and a rational* $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$*, and outputs a basis* $\mathbf{C}$ *of* L *satisfying* $\|\widetilde{\mathbf{C}}\| \leq \sqrt{n}\sigma$ *with probability* $\geq 1 - 2^{-\Omega(n)}$*. Further, the distribution of* $\mathbf{C}$ *is independent of the input basis* $\mathbf{B}$*.*

Let $m \geq n \geq 1$ and $q \geq 2$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, we define the lattice $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A} = \mathbf{0} \bmod q\}$. We will use an algorithm that jointly samples a uniform $\mathbf{A}$ and a short basis of $\Lambda_q^{\perp}(\mathbf{A})$.

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

Lemma 4 is often combined with the sampler from Lemma 2. Micciancio and Peikert [38] recently proposed a more efficient approach for this combined task, which should be preferred in practice but, for the sake of simplicity, we present our schemes using TrapGen. Lemma 4 was later extended by Gordon *et al.* [24] so that the columns of $\mathbf{A}$ lie within a prescribed linear vector subspace of $\mathbb{Z}_q^n$ (for q prime). For the security proof of our fully anonymous scheme, we will use an extension where the columns of the sampled $\mathbf{A}$ lie within a prescribed *affine* subspace of $\mathbb{Z}_q^n$. A proof is given in [31, Appendix C].

Lemma 5. *There exists a* PPT *algorithm* SuperSamp *that takes as inputs integers* $m \geq n \geq 1$ *and* $q \geq 2$ *prime such that* $m \geq \Omega(n \log q)$*, as well as matrices* $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$ *and* $\mathbf{C} \in \mathbb{Z}_q^{n \times n}$ *such that the rows of* $\mathbf{B}$ *span* $\mathbb{Z}_q^n$*. It outputs* $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{m \times n})$ *conditioned on* $\mathbf{B}^T \cdot \mathbf{A} = \mathbf{C}$*, and* $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{mn \log q \log m})$*.*

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

Lemma 6 ([17, Le. 3.2]). *There exists a* PPT *algorithm* ExtBasis *that takes as inputs a matrix* $\mathbf{B} \in \mathbb{Z}_q^{m' \times n}$ *whose first* m *rows span* $\mathbb{Z}_q^n$*, and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *where* $\mathbf{A}$ *is the top* $m \times n$ *submatrix of* $\mathbf{B}$*, and outputs a basis* $\mathbf{T_B}$ *of* $\Lambda_q^{\perp}(\mathbf{B})$ *with* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$*.*

For the sake of simplicity, we will assume that when the parameter conditions are satisfied, the distributions of the outputs of $\mathsf{TrapGen}$ and $\mathsf{SuperSamp}$ are *exactly* those they are meant to approximate, and the probabilistic norm bounds of Lemmas 1 and 3 *always* hold.

2.2 Computational problems

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

Definition 1. *Let m, q, β be functions of a parameter n. The Short Integer Solution problem $\mathsf{SIS}_{m,q,\beta}$ is as follows: Given $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, find $\mathbf{x} \in \Lambda_q^{\perp}(\mathbf{A})$ with $0 < \|\mathbf{x}\| \leq \beta$.*

Definition 2. *Let q, α be functions of a parameter n. For $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution $A_{q,\alpha,\mathbf{s}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by sampling $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ and (a noise) $e \hookleftarrow D_{\mathbb{Z},\alpha q}$, and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$. The Learning With Errors problem $\mathsf{LWE}_{q,\alpha}$ is as follows: For $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, distinguish between arbitrarily many independent samples from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ and the same number of independent samples from $A_{q,\alpha,\mathbf{s}}$.*

If $q \geq \sqrt{n}\beta$ and $m, \beta \leq \mathsf{poly}(n)$, then standard worst-case lattice problems with approximation factors $\gamma = \widetilde{\mathcal{O}}(\beta\sqrt{n})$ reduce to $\mathsf{SIS}_{m,q,\beta}$ (see, e.g., [23, Se. 9]). Similarly, if $\alpha q = \Omega(\sqrt{n})$, then standard worst-case lattice problems with approximation factors $\gamma = \mathcal{O}(\alpha/n)$ quantumly reduce to $\mathsf{LWE}_{q,\alpha}$ (see [43], and also [40, 11] for partial dequantizations). Note that we use the discrete noise variant of LWE from [24].

We will make use of a non-interactive zero-knowledge proof of knowledge (NIZPoK) protocol, which can be rather directly derived from [32, 34], for the following relation corresponding to an inhomogenous variant of the SIS relation:

$$R_{\mathsf{ISIS}} = \left\{(\mathbf{A}, \mathbf{y}, \beta; \mathbf{x}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^n \times \mathbb{Q} \times \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A} = \mathbf{y}^T \ \wedge \ \|\mathbf{x}\| \leq \beta\right\}.$$

The protocol, detailed in Section 2.3, is derived from the parallel repetition of a Σ-protocol with binary challenges. We call $\mathsf{Prove}_{\mathsf{ISIS}}$ and $\mathsf{Verify}_{\mathsf{ISIS}}$ the PPT algorithms run by the Prover and the Verifier when the scheme is rendered non-interactive using the Fiat-Shamir heuristic (*i.e.*, the challenge is implemented using the random oracle $H(\cdot)$). Algorithm $\mathsf{Prove}_{\mathsf{ISIS}}$ takes $(\mathbf{A}, \mathbf{y}, \beta; \mathbf{x})$ as inputs, and generates a transcript $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$. Algorithm $\mathsf{Verify}_{\mathsf{ISIS}}$ takes $(\mathbf{A}, \mathbf{y}, \beta)$ and such a transcript as inputs, and returns 0 or 1. The scheme has completeness error $2^{-\Omega(n)}$: if $\mathsf{Prove}_{\mathsf{ISIS}}$ is given as input an element of R_{ISIS}, then given as input the output of $\mathsf{Prove}_{\mathsf{ISIS}}$, $\mathsf{Verify}_{\mathsf{ISIS}}$ replies 1 with probability $\geq 1 - 2^{-\Omega(m)}$ (over the randomness of Prove). Also, there exists a PPT algorithm $\mathsf{Simulate}_{\mathsf{ISIS}}$ that, by reprogramming the random oracle $H(\cdot)$, takes $(\mathbf{A}, \mathbf{y}, \beta)$ as input and generates a transcript $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$ whose distribution is within statistical distance $2^{-\Omega(m)}$ of the genuine transcript distribution. Finally, there also exists a PPT algorithm $\mathsf{Extract}_{\mathsf{ISIS}}$ that given access to a time T algorithm $\mathcal{A}$ that generates transcripts accepted by $\mathsf{Verify}_{\mathsf{ISIS}}$ with probability ε, produces, in time $\mathrm{Poly}(T, 1/\varepsilon)$ a vector $\mathbf{x}'$ such that $(\mathbf{A}, \mathbf{y}, \mathcal{O}(\beta \cdot m^2); \mathbf{x}') \in R_{\mathsf{ISIS}}$.

We will also need a NIZKPoK protocol for the following language:

$$R_{\mathsf{LWE}} = \left\{ (\mathbf{A}, \mathbf{b}, \alpha; \mathbf{s}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \times \mathbb{Q} \times \mathbb{Z}_q^n : \|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}\| \leq \alpha q \sqrt{m} \right\}.$$

As noted in [34], we may multiply $\mathbf{b}$ by a parity check matrix $\mathbf{G} \in \mathbb{Z}_q^{(m-n) \times m}$ of $\mathbf{A}$ and prove the existence of small $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{e}^T \cdot \mathbf{G}^T = \mathbf{b}^T \cdot \mathbf{G}^T$. This may be done with the above NIZKPoK protocol for R_{ISIS}. We call $\mathsf{Prove}_{\mathsf{LWE}}$, $\mathsf{Verify}_{\mathsf{LWE}}$, $\mathsf{Simulate}_{\mathsf{LWE}}$ and $\mathsf{Extract}_{\mathsf{LWE}}$ the obtained PPT algorithms.

2.3 Proof of Knowledge of an ISIS Solution

In [32], Lyubashevsky described an identification scheme whose security relies on the hardness of the SIS problem. Given a public vector $\mathbf{y} \in \mathbb{Z}_q^n$ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the prover holds a short secret $\mathbf{x}$ and generates an interactive witness indistinguishable proof of knowledge of a short vector $\mathbf{x}'^T \in \mathbb{Z}^m$ such that $\mathbf{x}'^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$. A variant was later proposed in [34], which enjoys the property of being zero-knowledge (when the distribution of the transcript is conditioned on the prover not aborting). We present an adaptation of [34, Fig. 1] (still enjoying the same zero-knowledgedness property): the secret is a single vector, the challenges are binary (which we use for the extraction vector), and we increase the standard deviation of the commited vector to lower the rejection probability (we use a parallel repetition of the basic scheme, and want the probability that there is a reject among all the parallel iterations to be sufficiently away from 1).

Assume the prover P wishes to prove knowledge of an $\mathbf{x}$ such that $\mathbf{y}^T = \mathbf{x}^T \cdot \mathbf{A} \bmod q$ and $\|\mathbf{x}\| \leq \beta$, where $\mathbf{y}$ and $\mathbf{A}$ are public. The protocol takes place between the prover P and the verifier V and proceeds by the parallel repetition of a basic Σ-protocol with binary challenges. We set $\sigma = \Theta(\beta m^{3/2})$ and M_L as specified by [34, Th. 4.6]. Thanks to our larger value of σ, we obtain (by adapting [34, Le. 4.5]) that M_L is now $1 - \Omega(1/m)$.

1. The prover P generates a commitment $\mathsf{Comm} = (\mathbf{w}_i)_{i \leq t}$ where, for each $i \leq t$, $\mathbf{w}_i \in \mathbb{Z}_q^n$ is obtained by sampling $\mathbf{y}_i \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ and computing $\mathbf{w}_i^T = \mathbf{y}_i^T \cdot \mathbf{A} \bmod q$. The message Comm is sent to V.
2. The verifier V sends a challenge $\mathsf{Chall} \hookleftarrow \{0, 1\}^t$ to P.
3. For $i \leq t$, the prover P does the following.

   a. Compute $\mathbf{z}_i = \mathbf{y}_i + \mathsf{Chall}[i] \cdot \mathbf{x}$, where $\mathsf{Chall}[i]$ denotes the ith bit of Chall.
   b. Set $\mathbf{z}_i$ to $\perp$ with probability $\min(1, \frac{\exp(-\pi \|\mathbf{z}\|^2/\sigma^2)}{M_L \cdot \exp(-\pi \|\mathsf{Chall}[i] \cdot \mathbf{x} - \mathbf{z}\|^2/\sigma^2)})$.

   Then P sends the response $\mathsf{Resp} = (\mathbf{z}_i)_{i \leq t}$ to V.
4. The verifier V checks the transcript $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$ as follows:

a. For $i \leq t$, set $d_i = 1$ if $\|\mathbf{z}_i\| \leq 2\sigma\sqrt{m}$ and $\mathbf{z}_i^T \cdot \mathbf{A} = \mathbf{w}_i^T + \mathsf{Chall}[i] \cdot \mathbf{y}^T$. Otherwise, set $d_i = 0$.
b. Return 1 (and accept the transcript) if and only if $\sum_{i \leq t} d_i \geq 0.65t$.

The protocol has completeness error $2^{-\Omega(t)}$. Further, by [34, Th. 4.6], the distribution of the transcript conditioned on $\mathbf{z}_i \neq \perp$ can be simulated efficiently. Note that if we implement the challenge phase with a random oracle, we can compute the $\mathbf{z}_i$'s for increasing value of i, and repeat the whole procedure if $\mathbf{z}_i = \perp$ for some i. Thanks to our choice of σ, for any $t \leq \mathcal{O}(m)$, the probability that $\mathbf{z}_i = \perp$ for some i is $\leq c$, for some constant $c < 1$. Thanks to this random oracle enabled rejection, the simulator produces a distribution that is within statistical distance $2^{-\Omega(m)}$ to the transcript distribution.

Finally, the modified protocol provides special soundness in that there is a simple extractor that takes as input two valid transcripts $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$, $(\mathsf{Comm}, \mathsf{Chall}', \mathsf{Resp}')$ with distinct challenges $\mathsf{Chall} \neq \mathsf{Chall}'$ and obtains a witness $\mathbf{x}'$ such that $\mathbf{x}'^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$ and $\|\mathbf{x}'\| \leq \mathcal{O}(\sigma\sqrt{m}) \leq \mathcal{O}(\beta m^2)$.

2.4 Group Signatures

This section recalls the model of Bellare, Micciancio and Warinschi [5], which assumes static groups. A group signature scheme $\mathcal{GS}$ consists of a tuple of four PPT algorithms $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ with the following specifications:

- Keygen takes 1^n and 1^N as inputs, where $n \in \mathbb{N}$ is the security parameter, and $N \in \mathbb{N}$ is the maximum number of group members. It returns a tuple $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$ where gpk is the *group public key*, gmsk is the group manager secret key, and $\mathbf{gsk}$ is an N-vector of secret keys: $\mathsf{gsk}[j]$ is the signing key of the j-th user, for $j \in \{0, \ldots, N-1\}$.
- Sign takes the group public key gpk, a signing key $\mathsf{gsk}[j]$ and a message $M \in \{0,1\}^*$ as inputs. Its output is a signature $\Sigma \in \{0,1\}^*$ on M.
- Verify is deterministic and takes the group public key gpk, a message M and a putative signature Σ of M as inputs. It outputs either 0 or 1.
- Open is deterministic and takes as inputs the group public key gpk, the group manager secret key gmsk, a message M and a valid group signature Σ w.r.t. gpk. It returns an index $j \in \{0, \ldots, N-1\}$ or a special symbol $\perp$ in case of opening failure.

The group signature scheme must be *correct*, *i.e.*, for all integers n and N, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$ obtained from Keygen with $(1^n, 1^N)$ as input, all indexes $j \in \{0, \ldots, N-1\}$ and $M \in \{0,1\}^*$: $\mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = 1$ and $\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = j$, with probability negligibly close to 1 over the internal randomness of Keygen and Sign.

Bellare *et al.* [5] gave a unified security model for group signatures in static groups. The two main security requirements are *traceability* and *anonymity*. The former asks that no coalition of group members be able to create a signature that cannot be traced to one of them. The latter implies that, even if all group members' private keys are given to the adversary, signatures generated by two distinct members should be computationally indistinguishable.

```
$\underline{\mathbf{Exp}^{\text{anon-b}}_{\mathcal{GS},\mathcal{A}}(n,N)}$
$(\mathsf{gpk},\mathsf{gmsk},\mathbf{gsk}) \leftarrow \mathsf{Keygen}(1^n,1^N)$
$(\mathtt{st},j_0,j_1,M) \leftarrow \mathcal{A}(choose,\mathsf{gpk},\mathbf{gsk})$
$\Sigma^\star \leftarrow \mathsf{Sign}(\mathsf{gpk},\mathsf{gsk}[j_b],M)$
$b' \leftarrow \mathcal{A}(guess,\mathtt{st},\Sigma^\star)$
Return $b'$
```

```
$\underline{\mathbf{Exp}^{\text{trace}}_{\mathcal{GS},\mathcal{A}}(n,N)}$
$(\mathsf{gpk},\mathsf{gmsk},\mathbf{gsk}) \leftarrow \mathsf{Keygen}(1^n,1^N)$
$\mathtt{st} \leftarrow (\mathsf{gmsk},\mathsf{gpk})$
$\mathcal{C} \leftarrow \emptyset$ ; $K \leftarrow \varepsilon$ ; $Cont \leftarrow \mathtt{true}$
while $(Cont = \mathtt{true})$ do
  $(Cont,\mathtt{st},j) \leftarrow \mathcal{A}^{\mathcal{GS}.\mathsf{Sign}(\mathsf{gsk}[\cdot],\cdot)}(choose,\mathtt{st},K)$
  if $Cont = \mathtt{true}$ then $\mathcal{C} \leftarrow \mathcal{C} \cup \{j\}$;
    $K \leftarrow \mathsf{gsk}[j]$
  end if
end while;
$(M^\star,\Sigma^\star) \leftarrow \mathcal{A}^{\mathcal{GS}.\mathsf{Sign}(\mathsf{gsk}[\cdot],\cdot)}(guess,\mathtt{st})$
if $\mathsf{Verify}(\mathsf{gpk},M^\star,\Sigma^\star) = 0$ then Return 0
if $\mathsf{Open}(\mathsf{gmsk},M^\star,\Sigma^\star) = \perp$ then Return 1
if $\exists j^\star \in \{0,\ldots,N-1\}$ such that
        $(\mathsf{Open}(\mathsf{gmsk},M^\star,\Sigma^\star) = j^\star) \wedge (j^\star \notin \mathcal{C})$
        $\wedge ((j^\star,M^\star)$ not queried by $\mathcal{A})$
then Return 1 else Return 0
```

Fig. 1. Random experiments for anonymity and full traceability

Anonymity. Anonymity requires that, without the group manager's secret key, an adversary cannot recognize the identity of a user given its signature. More formally, the attacker, modeled as a two-stage adversary (*choose* and *guess*), is engaged in the first random experiment depicted in Figure 1. The *advantage* of such an adversary $\mathcal{A}$ against a group signature $\mathcal{GS}$ with N members is defined as $\mathbf{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n,N) = \left|\Pr[\mathbf{Exp}^{\text{anon-1}}_{\mathcal{GS},\mathcal{A}}(n,N)=1] - \Pr[\mathbf{Exp}^{\text{anon-0}}_{\mathcal{GS},\mathcal{A}}(n,N)=1]\right|$.

In our first scheme, we consider a *weak anonymity* scenario in which the adversary is not allowed to query an opening oracle. This relaxed model is precisely the one considered in [24], and was firstly introduced in [7]. Nonetheless, we provide in Section 5 a variant of our scheme enjoying chosen-ciphertext security. The adversary is then granted an access to an opening oracle that can be called on any string except the challenge signature $\Sigma^\star$.

Definition 3 (Weak and full anonymity, [5, 7]). *A group signature scheme $\mathcal{GS}$ is said to be* weakly anonymous *(resp.* fully anonymous*) if for all polynomial $N(\cdot)$ and all* PPT *adversaries $\mathcal{A}$ (resp.* PPT *adversaries $\mathcal{A}$ with access to an opening oracle which cannot be queried for the challenge signature), $\mathbf{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n,N)$ is a negligible function in the security parameter n.*

Full traceability. Full traceability ensures that all signatures, even those created by a coalition of users *and* the group manager, pooling their secret keys together, can be traced to a member of the forging coalition. Once again, the attacker is modeled as a two-stage adversary who is run within the second experiment described in Figure 1. Its success probability against $\mathcal{GS}$ is defined as $\mathbf{Succ}^{\text{trace}}_{\mathcal{GS},\mathcal{A}}(n,N) = \Pr[\mathbf{Exp}^{\text{trace}}_{\mathcal{GS},\mathcal{A}}(n,N)=1]$.

Definition 4 (Full traceability, [5]). *A group signature scheme $\mathcal{GS}$ is said to be* fully traceable *if for all polynomial $N(\cdot)$ and all* PPT *adversaries $\mathcal{A}$, its success probability $\boldsymbol{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$ is negligible in the security parameter n.*

3 An Asymptotically Shorter Lattice-Based Group Signature

At a high level, our key generation is based on the variant of Boyen's lattice signatures [8] described in [38, Se. 6.2]: Boyen's secret and verification keys respectively become our secret and public keys, whereas Boyen's message space is mapped to the users' identity space. There are however several additional twists in Keygen. First, each group member is given a *full* short basis of the public lattice associated to its identity, instead of a single short lattice vector. The reason is that, for anonymity and unlinkability purposes, the user has to generate each group signature using a *fresh* short lattice vector. Second, we sample our public key matrices $(\mathbf{A}_i)_{i\leq\ell}$ orthogonally to publicly known matrices $\mathbf{B}_i$, similarly to the group signature scheme from [24]. These $\mathbf{B}_i$'s will be used to publicly verify the validity of the signatures. They are sampled along with short trapdoor bases, using algorithm SuperSamp, which become part of the group signature secret key. These trapdoor bases will be used by the group authority to open signatures.

To anonymously sign M, the user samples a Boyen signature $(\mathbf{x}_1, \mathbf{x}_2)$ with its identity as message, which is a temporary certificate of its group membership. It does so using its full trapdoor matrix for the corresponding lattice. The user then encrypts $\mathbf{x}_2$, in a fashion that resembles [24], using Regev's dual encryption scheme from [23, Se. 7.1] with the $\mathbf{B}_i$'s as encryption public keys. Note that in all cases but one ($\mathbf{c}_0$ at Step 2), the signature is not embedded in the encryption noise as in [24], but as proper plaintext. The rest of the signing procedure consists in proving in zero-knowledge that these are valid ciphertexts and that the underlying plaintexts indeed encode a Boyen signature under the group public key. These ZKPoKs are all based on the interactive proof systems recalled in Sections 2.2 and 2.3. These were made non-interactive via the Fiat-Shamir heuristic with random oracle $H(\cdot)$ taking values in $\{0,1\}^t$. The message M is embedded in the application of the Fiat-Shamir transform at Step 6 of the signing algorithm.

The verification algorithm merely consists in verifying all proofs of knowledge concerning the Boyen signature embedded in the plaintexts of the ciphertexts.

Finally, the group manager can open any signature by decrypting the ciphertexts (using the group manager secret key) and then recovering the underlying Boyen signature within the plaintexts: this reveals which public key matrices $\mathbf{A}_i$ have been considered by the signer, and therefore its identity.

The scheme depends on several functions m, q, p, α and σ of the security parameter n and the group size $N(=2^\ell)$. They are set so that all algorithms can be implemented in polynomial time and are correct (Theorem 1), and so that the security properties (Theorems 2 and 3) hold, in the ROM, under the SIS and LWE hardness assumptions for parameters for which these problems enjoy reductions from standard worst-case lattice problems with polynomial approximation factors. More precisely, we require that:

- parameter m is $\Omega(n \log q)$,
- parameter σ is $\Omega(m^{3/2}\sqrt{\ell n \log q}\log m)$ and $\leq n^{\mathcal{O}(1)}$,
- parameter p is $\Omega((\alpha q + \sigma)m^{5/2})$,
- parameter α is set so that $\alpha^{-1} \geq \Omega(pm^3 \log m)$ and $\leq n^{\mathcal{O}(1)}$,
- parameter q is prime and $\Omega(\ell + \alpha^{-1}\sqrt{n\ell})$ and $\leq n^{\mathcal{O}(1)}$.

For example, we may set $m = \widetilde{\mathcal{O}}(n)$, $\sigma = \widetilde{\mathcal{O}}(n^2\sqrt{\ell})$, $p = \widetilde{\mathcal{O}}(n^{9/2}\sqrt{\ell})$ as well as $\alpha^{-1} = \widetilde{\mathcal{O}}(n^{15/2}\sqrt{\ell})$ and $q = \widetilde{\mathcal{O}}(\ell + n^8\sqrt{\ell})$.

Keygen$(1^n, 1^N)$**:** Given a security parameter $n > 0$ and the desired number of group members $N = 2^\ell \in \mathsf{poly}(n)$, choose parameters q, m, p, α and σ as specified above and make them public. Choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$ for some $t \in [\Omega(n), n^{\mathcal{O}(1)}]$, which will be modeled as a random oracle. Then, proceed as follows.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$.
2. For $i = 0$ to ℓ, sample $\mathbf{A}_i \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and compute $(\mathbf{B}_i, \mathbf{S}'_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}_i, \mathbf{0})$. Then, randomize $\mathbf{S}'_i$ as $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}'_i, \Omega(\sqrt{mn\log q}\log m))$.[6]
3. For $j = 0$ to $N-1$, let $\mathrm{id}_j = \mathrm{id}_j[1]\ldots\mathrm{id}_j[\ell] \in \{0,1\}^\ell$ be the binary representation of id_j and define: $\mathbf{A}_{\mathrm{id}_j} = \left[\frac{\mathbf{A}}{\mathbf{A}_0 + \sum_{i=1}^{\ell}\mathrm{id}_j[i]\mathbf{A}_i}\right] \in \mathbb{Z}_q^{2m\times n}$. Then, run $\mathbf{T}'_{\mathrm{id}_j} \leftarrow \mathsf{ExtBasis}(\mathbf{A}_{\mathrm{id}_j}, \mathbf{T_A})$ to get a short delegated basis $\mathbf{T}'_{\mathrm{id}_j}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$. Finally, run $\mathbf{T}_{\mathrm{id}_j} \leftarrow \mathsf{RandBasis}(\mathbf{T}'_{\mathrm{id}_j}, \Omega(m\sqrt{\ell n \log q \log m}))$.[6] The j-th member's private key is $\mathsf{gsk}[j] := \mathbf{T}_{\mathrm{id}_j}$.
4. The group manager's private key is $\mathsf{gmsk} := \{\mathbf{S}_i\}_{i=0}^{\ell}$ and the group public key is defined to be $\mathsf{gpk} := \big(\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_{i=0}^{\ell}\big)$. The algorithm outputs $\big(\mathsf{gpk}, \mathsf{gmsk}, \{\mathsf{gsk}[j]\}_{j=0}^{N-1}\big)$.

Sign$(\mathsf{gpk}, \mathsf{gsk}[j], M)$**:** To sign a message $M \in \{0,1\}^*$ using the private key $\mathsf{gsk}[j] = \mathbf{T}_{\mathrm{id}_j}$, proceed as follows.

1. Run $\mathsf{GPVSample}(\mathbf{T}_{\mathrm{id}_j}, \sigma)$ to get $(\mathbf{x}_1^T|\mathbf{x}_2^T)^T \in \Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$ of norm $\leq \sigma\sqrt{2m}$.
2. Sample $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and encrypt $\mathbf{x}_2 \in \mathbb{Z}_q^m$ as $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$.
3. Sample $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. For $i = 1$ to ℓ, sample $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_j[i] \cdot \mathbf{x}_2$, which encrypts $\mathbf{x}_2 \in \mathbb{Z}_q^m$ (resp. $\mathbf{0}$) if $\mathrm{id}_j[i] = 1$ (resp. $\mathrm{id}_j[i] = 0$).
4. Generate a NIZKPoK π_0 of $\mathbf{s}_0$ so that $(\mathbf{B}_0, \mathbf{c}_0, \sqrt{2}\sigma/q; \mathbf{s}_0) \in R_{\mathsf{LWE}}$ (see Section 2.2).
5. For $i = 1$ to ℓ, generate a NIZKPoK $\pi_{\mathrm{OR},i}$ of $\mathbf{s}$ and $\mathbf{s}_0$ so that either:
   (i) $((\mathbf{B}_i|\mathbf{B}_0), p^{-1}(\mathbf{c}_i - \mathbf{c}_0), \sqrt{2}\alpha; (\mathbf{s}^T| - \mathbf{s}_0^T)^T) \in R_{\mathsf{LWE}}$ (the vectors $\mathbf{c}_i$ and $\mathbf{c}_0$ encrypt the same $\mathbf{x}_2$, so that $p^{-1}(\mathbf{c}_i - \mathbf{c}_0)$ is close to the $\mathbb{Z}_q$-span of $(\mathbf{B}_i|\mathbf{B}_0)$);

[6] These randomisation steps are not needed for the correctness of the scheme but are important in the traceability proof.

(ii) or $(\mathbf{B}_i, p^{-1}\mathbf{c}_i, \alpha; \mathbf{s}) \in R_{\mathsf{LWE}}$ (the vector $\mathbf{c}_i$ encrypts $\mathbf{0}$, so that $p^{-1}\mathbf{c}_i$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_i$).

This can be achieved by OR-ing two proofs for R_{LWE}, and making the resulting protocol non-interactive with the Fiat-Shamir heuristic.[7]

6. For $i = 1$ to ℓ, set $\mathbf{y}_i = \mathrm{id}_j[i]\mathbf{x}_2 \in \mathbb{Z}^m$ and generate a NIZKPoK π_K of $(\mathbf{e}_i)_{1\le i\le \ell}, (\mathbf{y}_i)_{1\le i\le \ell}, \mathbf{x}_1$ such that, for $i \in [1,\ell]$

$$\mathbf{x}_1^T\mathbf{A} + \sum_{i=0}^{\ell} \mathbf{c}_i^T \mathbf{A}_i = \sum_{i=1}^{\ell} \mathbf{e}_i^T \big(p\mathbf{A}_i\big) \quad \text{and} \quad \mathbf{e}_i^T\big(p\mathbf{A}_i\big) + \mathbf{y}_i^T\mathbf{A}_i = \mathbf{c}_i^T\mathbf{A}_i \tag{2}$$

with $\|\mathbf{e}_i\|, \|\mathbf{y}_i\|, \|\mathbf{x}_1\| \le \max(\sigma, \alpha q)\sqrt{m}$ for all i. This is achieved using $\mathsf{Prove}_{\mathsf{ISIS}}$ in order to produce a triple $(\mathsf{Comm}_K, \mathsf{Chall}_K, \mathsf{Resp}_K)$, where $\mathsf{Chall}_K = H(M, \mathsf{Comm}_K, (\mathbf{c}_i)_{0\le i\le\ell}, \pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le \ell})$.

The signature consists of

$$\Sigma = \big((\mathbf{c}_i)_{0\le i\le \ell}, \pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le\ell}, \pi_K\big). \tag{3}$$

Verify$(\mathsf{gpk}, M, \Sigma)$**:** Parse Σ as in (3). Then, return 1 if $\pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le\ell}, \pi_K$ properly verify. Else, return 0.

Open$(\mathsf{gpk}, \mathsf{gmsk}, M, \Sigma)$**:** Parse gmsk as $\{\mathbf{S}_i\}_{i=0}^{\ell}$ and Σ as in (3). Compute $\mathbf{x}_2$ by decrypting $\mathbf{c}_0$ using $\mathbf{S}_0$. For $i = 1$ to ℓ, use $\mathbf{S}_i$ to determine which one of the vectors $p^{-1}\mathbf{c}_i$ and $p^{-1}(\mathbf{c}_i - \mathbf{x}_2)$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_i$. Set $\mathrm{id}[i] = 0$ in the former case and $\mathrm{id}[i] = 1$ in the latter. Eventually, output $\mathrm{id} = \mathrm{id}[1]\ldots\mathrm{id}[\ell]$.

All steps of the scheme above can be implemented in polynomial-time as a function of the security parameter n, assuming that $q \ge 2$ is prime, $m \ge \Omega(n\log q)$, $\sigma \ge \Omega(m^{3/2}\sqrt{\ell n \log q}\log m)$ (using Lemmas 2 and 3), and $\alpha q \ge \Omega(1)$ (using Lemma 2). Under some mild conditions on the parameters, the scheme above is correct, *i.e.*, the verifier accepts honestly generated signatures, and the group manager successfully opens honestly generated signatures. In particular, multiplying the ciphertexts by the $\mathbf{S}_i$ modulo q should reveal $p\cdot\mathbf{e}_i + \mathrm{id}_j[i]\cdot\mathbf{x}_2$ over the integers, and $\|\mathrm{id}_j[i]\cdot\mathbf{x}_2\|_\infty$ should be smaller than p.

Theorem 1. *Let us assume that $q \ge 2$ is prime and that we have $m \ge \Omega(n\log q)$, $\sigma \ge \Omega(m^{3/2}\sqrt{\ell n\log q}\log m)$, $\alpha^{-1} \ge \Omega(pm^{5/2}\log m\sqrt{n\log q})$ as well as $q \ge \Omega(\alpha^{-1} + \sigma m^{5/2}\log m\sqrt{n\log q})$. Then, the group signature scheme above can be implemented in time polynomial in n, is correct, and the bit-size of the generated signatures in $\mathcal{O}(\ell t m \log q)$.*

4 Security

We now focus on the security of the scheme of Section 3.

[7] The disjunction of two relations that can be proved by Σ-protocols can also be proved by a Σ-protocol [20, 21].

Anonymity. Like in [24, 7], we use a relaxation of the anonymity definition, called weak anonymity and recalled in Definition 3. Analogously to the notion of IND-CPA security for public-key encryption, the adversary does not have access to a signature opening oracle. We show that the two versions (for $b = 0, 1$) of the anonymity security experiment recalled in Figure 1 are indistinguishable under the LWE assumption. We use several intermediate hybrid experiments called $G_b^{(i)}$, and show that each of these experiments is indistinguishable from the next one. At each step, we only change one element of the game (highlighted by an arrow in Figure 2), to finally reach the experiment $G^{(4)}$ where the signature scheme does not depend on the identity of the user anymore.

Theorem 2. *In the random oracle model, the scheme provides weak anonymity in the sense of Definition 3 under the* $\mathsf{LWE}_{q,\alpha}$ *assumption. Namely, for any* PPT *adversary* $\mathcal{A}$ *with advantage* ε*, there exists an algorithm* $\mathcal{B}$ *solving the* $\mathsf{LWE}_{q,\alpha}$ *problem with the same advantage.*

Proof. We define by G_0 the experiment of Definition 3 with $b = 0$ and by G_1 the same experiment with $b = 1$. To show the anonymity of the scheme, we prove that G_0 and G_1 are indistinguishable. We use several hybrid experiments named $\mathrm{G}_b^{(1)}$, $\mathrm{G}_b^{(2)}$, $\mathrm{G}_b^{(3)}$ and $G^{(4)}$ (described in Figure 2), where b is either 0 or 1.

Lemma 7. *For each* $b \in \{0, 1\}$*,* G_b *and* $G_b^{(1)}$ *are statistically indistinguishable.*

We only change the way we generate $(\mathbf{x}_1^T|\mathbf{x}_2^T)^T$, by using the fact that one way to generate it is to first sample $\mathbf{x}_2$ from $D_{\mathbb{Z}^m,\sigma}$ and then generate $\mathbf{x}_1$ from $D_{\mathbb{Z}^m,\sigma}$ such that $(\mathbf{x}_1^T|\mathbf{x}_2^T) \cdot \mathbf{A}_{id_{j_b}} = 0 \bmod q$ (by using the trapdoor $\mathbf{T}_\mathbf{A}$). This change is purely conceptual and the vector $(\mathbf{x}_1^T|\mathbf{x}_2^T)^T$ has the same distribution anyway. The two experiments are thus identical from $\mathcal{A}$'s view and $\mathbf{x}_2$ is chosen independently of the signer's identity in the challenge phase.

Lemma 8. *For each* $b \in \{0, 1\}$*,* $G_b^{(1)}$ *and* $G_b^{(2)}$ *are statistically indistinguishable.*

The differences are simply: Instead of generating the proofs $\{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}$ and π_K using the witnesses, we simulate them (see Section 2.2).

Lemma 9. *For each* $b \in \{0, 1\}$*, if the* $\mathsf{LWE}_{q,\alpha}$ *problem is hard, then the experiments* $G_b^{(2)}$ *and* $G_b^{(3)}$ *are computationally indistinguishable.*

Proof. This proof uses the same principle as the proof of [24, Claim 1]: We use the adversary $\mathcal{A}$ to construct a PPT algorithm $\mathcal{B}$ for the $\mathsf{LWE}_{q,\alpha}$ problem. We consider an LWE instance $(\mathbf{B}', \mathbf{z}) \in \mathbb{Z}_q^{m\ell \times (n+1)}$ such that $\mathbf{B}' = (\mathbf{B}'_1, \ldots, \mathbf{B}'_\ell)$ and $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$ with $\mathbf{B}'_i \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{z}_i \in \mathbb{Z}_q^m$. The component $\mathbf{z}$ is either uniform in $\mathbb{Z}_q^{m\ell}$, or of the form $\mathbf{z} = \mathbf{B}' \cdot \mathbf{s} + \mathbf{e}$ where $\mathbf{e}$ is sampled from $D_{\mathbb{Z}^{m\ell},\alpha q}$.

We construct a modified Keygen algorithm using this LWE instance: It generates the matrix $\mathbf{A}$ with a basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$. Instead of generating the $\mathbf{B}_i$'s genuinely, we pick $\mathbf{B}_0$ uniformly in $\mathbb{Z}^{m \times n}$ and set $\mathbf{B}_i = \mathbf{B}'_i$ for $1 \le i \le \ell$. For $0 \le i \le \ell$, we compute $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_i, \mathbf{0})$. Then, for each

Experiment $\mathbf{G}_b$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  1. $(\mathbf{x}_1^T|\mathbf{x}_2^T)^T \hookleftarrow \mathsf{GPVSample}(\mathbf{T}_{\mathrm{id}_{j_b}}, \sigma)$;
  we have $(\mathbf{x}_1^T|\mathbf{x}_2^T) \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$, compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$.
  3. Choose $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, and for $i = 1$ to ℓ, choose $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  5. Generate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Generate π_K.

Experiment $\mathbf{G}_b^{(1)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  $\to$ **1.** Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and, using $\mathbf{T}_{\mathbf{A}}$, sample $\mathbf{x}_1 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ conditioned on $(\mathbf{x}_1^T|\mathbf{x}_2^T) \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$, compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  3. Choose $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, and for $i = 1$ to ℓ, choose $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  5. Generate $\{\pi_{OR,i}\}_i$.
  6. Generate π_K.

Experiment $\mathbf{G}_b^{(2)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  1. Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$; sample $\mathbf{x}_1 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, conditioned on
  $(\mathbf{x}_1^T|\mathbf{x}_2^T) \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  3. Choose $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, and for $i = 1$ to ℓ, choose $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  $\to$ **5.** Simulate $\{\pi_{\mathrm{OR},i}\}_i$.
  $\to$ **6.** Simulate π_K.

Experiment $\mathbf{G}_b^{(3)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  1. Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ Sample $\mathbf{x}_1 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ conditioned on
  $(\mathbf{x}_1^T|\mathbf{x}_2^T) \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  $\to$ **3.** For $i = 1$ to ℓ, choose $\mathbf{z}_i \hookleftarrow U(\mathbb{Z}_q^m)$ and compute $\mathbf{c}_i = \mathbf{z}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  5. Simulate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Simulate π_K.

Experiment $\mathbf{G}^{(4)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  $\to$ **1.** Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  $\to$ **3.** For $i = 1$ to ℓ, choose $\mathbf{z}_i \hookleftarrow U(\mathbb{Z}_q^m)$ and set $\mathbf{c}_i = \mathbf{z}_i$.
  4. Generate π_0.
  5. Simulate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Simulate π_K.

Fig. 2. Experiments G_b, $\mathrm{G}_b^{(1)}$, $\mathrm{G}_b^{(2)}$,$\mathrm{G}_b^{(3)}$ and $\mathrm{G}^{(4)}$.

$j \in [0, N-1]$, we define $\mathbf{A}_{\mathrm{id}_j}$ as in the original Keygen algorithm, and compute a trapdoor $\mathbf{T}_{id_j}$ using $\mathbf{T}_{\mathbf{A}}$. The adversary $\mathcal{A}$ is given gpk and $\{\mathsf{gsk}_j\}_j$. In the challenge phase, it outputs j_0, j_1 and a message M. By [24], this Keygen algorithm and the one in all the experiments are statistically indistinguishable. Then, the signature is created on behalf of the group member j_b. Namely, $\mathcal{B}$ first chooses $\mathbf{x}_2 \leftarrow D_{\mathbb{Z}^m,\sigma}$ and finds $\mathbf{x}_1$ such that $(\mathbf{x}_1^T|\mathbf{x}_2^T)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$. Then it chooses $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and computes $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$. Third, it computes

$\mathbf{c}_i = p \cdot \mathbf{z}_i + \mathrm{id}_{j_b}[i] \cdot x_2$ (with the $\mathbf{z}_i$ of the LWE instance). Then it generates π_0 and simulates the $\pi_{\mathrm{OR},i}$'s and π_K proofs.

We let $\mathcal{D}_{\mathsf{LWE}}$ denote this experiment when $\mathbf{z} = \mathbf{s} \cdot \mathbf{B}'^T + \mathbf{e}$: This experiment is statistically close to $\mathrm{G}_b^{(2)}$. Then, we let $\mathcal{D}_{rand}$ denote this experiment when $\mathbf{z}$ is uniform: It is statistically close to $\mathrm{G}_b^{(3)}$. As a consequence, if the adversary $\mathcal{A}$ can distinguish between the experiments $\mathrm{G}_b^{(2)}$ and $\mathrm{G}_b^{(3)}$ with non-negligible advantage, then we can solve the $\mathsf{LWE}_{q,\alpha}$ problem with the same advantage.

Lemma 10. *For each $b \in \{0,1\}$, $G_b^{(3)}$ and $G^{(4)}$ are indistinguishable.*

Between these two experiments, we change the first and third steps. In the former, we no longer generate $\mathbf{x}_1$ and, in the latter, $\mathbf{c}_i$ is uniformly sampled in $\mathbb{Z}_q^m$. These changes are purely conceptual. Indeed, in experiment $G_b^{(3)}$, $\mathbf{x}_1$ is not used beyond Step 1. In the same experiment, we also have $\mathbf{c}_i = \mathbf{z}_i + id_{j_b}[i]$. Since the $\mathbf{z}_i$'s are uniformly sampled in $\mathbb{Z}_q^m$, the $\mathbf{c}_i$'s are also uniformly distributed in $\mathbb{Z}_q^m$. As a consequence, the $\mathbf{c}_i$'s of $G_b^{(3)}$ and the $\mathbf{c}_i$'s of $G^{(4)}$ have the same distribution. In $G_b^{(4)}$, we conclude that $\mathcal{A}$'s view is exactly the same as in experiments $G_b^{(3)}$. Since the experiment $G^{(4)}$ no longer depends on the bit $b \in \{0,1\}$ that determines the signer's identity, the announced result follows. $\square$

Traceability. The proof of traceability relies on the technique of [1, 8] and a refinement from [28, 38], which is used in order to allow for a smaller modulus q.

A difference with the proof of [24] is that we need to rely on the knowledge extractor of a proof of knowledge π_K. Depending on whether the extracted witnesses $\{\mathbf{e}_i, \mathbf{y}_i\}_{i=1}^{\ell}$ of relation (2) satisfy $\mathbf{y}_i = \mathrm{id}_j[i]\mathbf{x}_2$ for all i or not, we need to distinguish two cases. The strategy of the reduction and the way it uses its given $\mathsf{SIS}_{q,\beta}$ instance will depend on which case is expected to occur. The proof is provided is the long version of this article [31, Section 4.2].

Theorem 3. *Assume that $q > \log N$, $p \geq \Omega((\alpha q + \sigma)m^{5/2})$ and $\beta \geq \Omega(\sigma m^{7/2}\sqrt{\log N} + p\alpha q m^{5/2})$. Then for any PPT traceability adversary $\mathcal{A}$ with success probability ε, there exists a PPT algorithm $\mathcal{B}$ solving $\mathsf{SIS}_{m,q,\beta}$ with probability $\varepsilon'' \geq \frac{\varepsilon'}{2N} \cdot (\frac{\varepsilon'}{q_H} - 2^{-t}) + \frac{\varepsilon'}{2\log N}$, where $\varepsilon' = \varepsilon - 2^{-t} - 2^{-\Omega(n)}$ and q_H is the number of queries to the random oracle $H : \{0,1\}^* \to \{0,1\}^t$.*

5 A Variant with Full (CCA-)Anonymity

We modify our basic group signature scheme to reach the strongest anonymity level (Definition 3), in which the attacker is authorized to query an opening oracle. This implies the simulation of an oracle which opens adversarially-chosen signatures in the proof of anonymity. To this end, we replace each $\mathbf{B}_i$ from our previous scheme by a matrix $\mathbf{B}_{i,\mathsf{VK}}$ that depends on the verification key VK of a strongly unforgeable one-time signature. The reduction will be able to compute a trapdoor for all these matrices, except for one specific verification key $\mathsf{VK}^\star$

that will be used in the challenge phase. This will provide the reduction with a backdoor allowing it to open all adversarially-generated signatures.

It is assumed that the one-time verification keys VK belong to $\mathbb{Z}_q^n$ (note that this condition can always be enforced by hashing VK). Following Agrawal *et al.* [1], we rely on a full-rank difference function $H_{vk} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ such that, for any two distinct $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, the difference $H_{vk}(\mathbf{u}) - H_{vk}(\mathbf{v})$ is a full rank matrix.

$\mathsf{Keygen}(1^n, 1^N)$: Given a security parameter $n > 0$ and the desired number of members $N = 2^\ell \in \mathsf{poly}(n)$, choose parameters as in Section 3 and make them public. Choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$, that will be seen as a random oracle, and a one-time signature $\Pi^{\mathrm{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$.
2. For $i = 0$ to ℓ, repeat the following steps.
   a. Choose uniformly random matrices $\mathbf{A}_{i,1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1} \in \mathbb{Z}_q^{m \times n}$.
   b. Compute $(\mathbf{A}_{i,2}, \mathbf{T}_{i,2}) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_{i,1}, 0^{n \times n})$ such that $\mathbf{B}_{i,1}^T \cdot \mathbf{A}_{i,2} = 0 \bmod q$ and discard $\mathbf{T}_{i,2}$, which will not be needed. Define $\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_{i,1} \\ \hline \mathbf{A}_{i,2} \end{bmatrix} \in \mathbb{Z}_q^{2m \times n}$.
   c. Run $(\mathbf{B}_{i,-1}, \mathbf{S}'_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}_{i,1}, -\mathbf{A}_{i,2}^T \cdot \mathbf{B}_{i,0})$ to obtain $\mathbf{B}_{i,-1} \in \mathbb{Z}_q^{m \times n}$ such that $\mathbf{B}_{i,-1}^T \cdot \mathbf{A}_{i,1} + \mathbf{B}_{i,0}^T \cdot \mathbf{A}_{i,2} = 0 \bmod q$.
   d. Compute a re-randomized trapdoor $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}'_i)$ for $\mathbf{B}_{i,-1}$.

   For any string VK, if the matrix $H_{vk}(\mathsf{VK})$ is used to define $\mathbf{B}_{i,\mathsf{VK}} = \begin{bmatrix} \mathbf{B}_{i,-1} \\ \hline \mathbf{B}_{i,0} + \mathbf{B}_{i,1} H_{vk}(\mathsf{VK}) \end{bmatrix} \in \mathbb{Z}_q^{2m \times n}$, we have $\mathbf{B}_{i,\mathsf{VK}}^T \cdot \mathbf{A}_i = 0 \bmod q$ for all i.
3. For $j = 0$ to $N-1$, let $\mathrm{id}_j = \mathrm{id}_j[1] \ldots \mathrm{id}_j[\ell] \in \{0,1\}^\ell$ be the binary representation of id_j and define: $\mathbf{A}_{\mathrm{id}_j} = \begin{bmatrix} \mathbf{A} \\ \hline \mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i] \mathbf{A}_i \end{bmatrix} \in \mathbb{Z}_q^{3m \times n}$. Then run $\mathbf{T}_{\mathrm{id}_j} \leftarrow \mathsf{ExtBasis}(\mathbf{T_A}, \mathbf{A}_{\mathrm{id}_j})$ to get a short delegated basis $\mathbf{T}_{\mathrm{id}_j} \in \mathbb{Z}^{3m \times 3m}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$ and define $\mathsf{gsk}[j] := \mathbf{T}_{\mathrm{id}_j}$.
4. Finally, define $\mathsf{gpk} := \big(\mathbf{A}, \{\mathbf{A}_i, (\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1})\}_{i=0}^{\ell},\ H,\ \Pi^{\mathrm{ots}}\big)$ and $\mathsf{gmsk} := \{\mathbf{S}_i\}_{i=0}^{\ell}$. The algorithm outputs $\big(\mathsf{gpk}, \mathsf{gmsk}, \{\mathsf{gsk}[j]\}_{j=0}^{N-1}\big)$.

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$: To sign a message $M \in \{0,1\}^*$ using the private key $\mathsf{gsk}[j] = \mathbf{T}_{\mathrm{id}_j}$, generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^n)$ for Π^{ots} and proceed as follows.

1. Run $\mathsf{GPVSample}(\mathbf{T}_{\mathrm{id}_j}, \sigma)$ to get $(\mathbf{x}_1^T | \mathbf{x}_2^T)^T \in \Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$ of norm $\leq \sigma\sqrt{3m}$.
2. Sample $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and encrypt $\mathbf{x}_2 \in \mathbb{Z}_q^{2m}$ as $\mathbf{c}_0 = \mathbf{B}_{0,\mathsf{VK}} \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^{2m}$.
3. Sample $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. For $i = 1$ to ℓ, sample $\mathbf{e}_i \leftarrow D_{\mathbb{Z}^m, \alpha q}$ and a random matrix $\mathbf{R}_i \in \mathbb{Z}^{m \times m}$ whose columns are sampled from $D_{\mathbb{Z}^m, \sigma}$. Then, compute $\mathbf{c}_i = \mathbf{B}_{i,\mathsf{VK}} \cdot \mathbf{s} + p \cdot \big[\mathbf{e}_i \,\big|\, \mathbf{e}_i \cdot \mathbf{R}_i\big] + \mathrm{id}_j[i] \cdot \mathbf{x}_2$, which encrypts $\mathbf{x}_2 \in \mathbb{Z}_q^{2m}$ (resp. 0^{2m}) if $\mathrm{id}_j[i] = 1$ (resp. $\mathrm{id}_j[i] = 0$).
4. Generate a NIZKPoK π_0 of $\mathbf{s}_0$ so that $(\mathbf{B}_0, \mathbf{c}_0, \sqrt{2}\sigma/q; \mathbf{s}_0) \in R_{\mathsf{LWE}}$.
5. For $i = 1$ to ℓ, generate a NIZKPoK $\pi_{\mathrm{OR},i}$ of $\mathbf{s}$ and $\mathbf{s}_0$ so that either:
   (i) $((\mathbf{B}_{i,\mathsf{VK}} | \mathbf{B}_{0,\mathsf{VK}}), p^{-1}(\mathbf{c}_i - \mathbf{c}_0), \sqrt{2}\alpha; (\mathbf{s}^T | -\mathbf{s}_0^T)^T) \in R_{\mathsf{LWE}}$ (the vectors $\mathbf{c}_i$ and $\mathbf{c}_0$ encrypt the same $\mathbf{x}_2$, so that the vector $p^{-1}(\mathbf{c}_i - \mathbf{c}_0)$ is close to the $\mathbb{Z}_q$-span of $(\mathbf{B}_{i,\mathsf{VK}} | \mathbf{B}_{0,\mathsf{VK}})$);

(ii) or $(\mathbf{B}_{i,\mathsf{VK}}^T, p^{-1}\mathbf{c}_i, \alpha; \mathbf{s}) \in R_{\mathsf{LWE}}$ ($p^{-1}\mathbf{c}_i$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_{i,\mathsf{VK}}^T$).

6. For $i = 1$ to ℓ, set $\mathbf{y}_i = \mathrm{id}_j[i]\mathbf{x}_2 \in \mathbb{Z}^{2m}$ and generate a NIZKPoK π_K of $(\mathbf{e}_i)_{1\le i\le \ell}$, $(\mathbf{y}_i)_{1\le i\le \ell}$, $\mathbf{x}_1$ such that: $\mathbf{x}_1^T\mathbf{A} + \sum_{i=0}^{\ell} \mathbf{c}_i^T\mathbf{A}_i = \sum_{i=1}^{\ell} \mathbf{e}_i^T\big(p\cdot \mathbf{A}_i\big)$ and $\mathbf{e}_i^T\big(p\cdot\mathbf{A}_i\big) + \mathbf{y}_i^T\mathbf{A}_i = \mathbf{c}_i^T\mathbf{A}_i$with $||\mathbf{x}_1|| \le \sigma\sqrt{m}$ and $||\mathbf{y}_i|| \le \sigma\sqrt{2m}$ for each $i \in \{1,\ldots,\ell\}$.
   This is achieved using $\mathsf{Prove}_{\mathsf{ISIS}}$, giving a triple $(\mathsf{Comm}_K, \mathsf{Chall}_K, \mathsf{Resp}_K)$, where $\mathsf{Chall}_K = H(M, \mathsf{Comm}_K, (\mathbf{c}_i)_{0\le i\le\ell}, \pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le\ell})$.
7. Compute $sig = \mathcal{S}(\mathsf{SK}, (\mathbf{c}_i)_{0\le i\le\ell}, \pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le\ell}, \pi_K))$.

The signature consists of

$$\varSigma = \big(\mathsf{VK}, \mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_\ell, \pi_0, \pi_{\mathrm{OR},1}, \ldots, \pi_{\mathrm{OR},\ell}, \pi_K, sig\big). \tag{4}$$

Verify$(\mathsf{gpk}, M, \varSigma)$**:** Parse the signature $\varSigma$ as in (4). Then, return 0 in the event that $\mathcal{V}(\mathsf{VK}, sig, (\mathbf{c}_i)_{0\le i\le\ell}, \pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le\ell}, \pi_K)) = 0$. Then, return 1 if all proofs $\pi_0, (\pi_{\mathrm{OR},i})_{1\le i\le\ell}, \pi_K$ properly verify. Otherwise, return 0.

Open$(\mathsf{gpk}, \mathsf{gmsk}, M, \varSigma)$**:** Parse gmsk as $\{\mathbf{S}_i\}_{i=0}^{\ell}$ and $\varSigma$ as in (4). For $i = 0$ to ℓ, compute a trapdoor $\mathbf{S}_{i,\mathsf{VK}} \leftarrow \mathsf{ExtBasis}(\mathbf{S}_i, \mathbf{B}_{i,\mathsf{VK}})$ for $\mathbf{B}_{i,\mathsf{VK}}$. Using the delegated basis $\mathbf{S}_{0,\mathsf{VK}} \in \mathbb{Z}^{2m\times 2m}$ (for which we have $\mathbf{S}_{0,\mathsf{VK}} \cdot \mathbf{B}_{0,\mathsf{VK}} = 0 \bmod q$), compute $\mathbf{x}_2$ by decrypting $\mathbf{c}_0$. Then, using $\mathbf{S}_{i,\mathsf{VK}} \in \mathbb{Z}^{2m\times 2m}$, determine which vector among $p^{-1}\mathbf{c}_i \bmod q$ and $p^{-1}(\mathbf{c}_i - \mathbf{x}_2) \bmod q$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_{i,\mathsf{VK}}$. Set $\mathrm{id}[i] = 0$ in the former case and $\mathrm{id}[i] = 1$ in the latter case. Eventually, output $\mathrm{id} = \mathrm{id}[1]\ldots\mathrm{id}[\ell]$.

In [31, Appendix D], we prove the following theorems.

Theorem 4. *The scheme provides full anonymity in the ROM if the* $\mathsf{LWE}_{q,\alpha}$ *assumption holds and if the one-time signature is strongly unforgeable.*

Theorem 5. *Assuming that* $q > \log N$, *the scheme is fully traceable in the ROM under the* $\mathsf{SIS}_{q,\beta}$ *assumption. More precisely, for any* PPT *traceability adversary* $\mathcal{A}$ *with success probability* ε, *there exists an algorithm* $\mathcal{B}$ *solving the* $\mathsf{SIS}_{q,\beta}$ *problem with probability at least* $\frac{1}{2N}\cdot\left(\varepsilon - \frac{1}{2^t}\right)\cdot\left(\frac{\varepsilon-1/2^t}{q_H} - \frac{1}{2^t}\right)$, *where* q_H *is the number of queries to* $H : \{0,1\}^* \to \{0,1\}^t$.

Acknowledgements. We thank Daniele Micciancio and Khoa Nguyen for helpful discussions. Parts of the research described in this work were underwent while the third author was visiting ENS de Lyon, under the INRIA invited researcher scheme. The last author was partly supported by the Australian Research Council Discovery Grant DP110100628.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of Eurocrypt*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010.

2. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
3. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proc. of Crypto*, number 1880 in LNCS, pages 255–270. Springer, 2000.
4. W. Banaszczyk. New bounds in some transference theorems in the geometry of number. *Math. Ann*, 296:625–635, 1993.
5. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Proc. of Eurocrypt*, volume 2656 of *LNCS*, pages 614–629, 2003.
6. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Proc. of CT-RSA*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.
7. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. of Crypto*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
8. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Proc. of PKC*, volume 6056 of *LNCS*, pages 499–517. Springer, 2010.
9. X. Boyen and B. Waters. Compact group signatures without random oracles. In *Proc. of Eurocrypt*, volume 4004 of *LNCS*, pages 427–444. Springer, 2006.
10. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Proc. of PKC*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.
11. Z. Brakerski, A. Langlois, C. Peikert, Regev. O., and D. Stehlé. On the classical hardness of learning with errors. In *Proc. of STOC*, pages 575–584. ACM, 2013.
12. E. Brickell. An efficient protocol for anonymously providing assurance of the container of a private key., 2003. Submitted to the Trusted Computing Group.
13. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Proc. of Crypto*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
14. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Proc. of SCN*, volume 2576 of *LNCS*, pages 268–289. Springer, 2002.
15. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Proc. of Crypto*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004.
16. J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *Proc. of SCN*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012.
17. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of Eurocrypt*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.
18. D. Chaum and E. van Heyst. Group signatures. In *Proc. of Eurocrypt*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
19. J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *Proc. of Crypto*, volume 8042 of *LNCS*, pages 476–493. Springer, 2013.
20. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proc. of Crypto*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.
21. I Damgård. On Σ-protocols. Manuscript, 2010. Available at `http://www.daimi.au.dk/~ivan/Sigma.pdf`.
22. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices and applications. In *Proc. of Eurocrypt*, LNCS. Springer, 2013.

23. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of STOC*, pages 197–206. ACM, 2008.
24. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Proc. of Asiacrypt*, volume 2647 of *LNCS*, pages 395–412. Springer, 2010.
25. J. Groth. Fully anonymous group signatures without random oracles. In *Proc. of Asiacrypt*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
26. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Proc. of Eurocrypt*, volume 4004 of *LNCS*, pages 339–358. Springer, 2006.
27. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Proc. of Eurocrypt*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.
28. S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *Proc. of Crypto*, volume 5677 of *LNCS*, pages 654–670. Springer, 2009.
29. VSC Project IEEE P1556 Working Group. Dedicated short range communications (dsrc), 2003.
30. A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *(IJSN)*, 1(1/2):24–45, 2006.
31. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. Cryptology ePrint Archive, Report 2013/308, 2013. `http://eprint.iacr.org/2013/308`.
32. V. Lyubashevsky. Lattice-based identification schemes secure unde active attacks. In *Proc. of PKC*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
33. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Proc. of Asiacrypt*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.
34. V. Lyubashevsky. Lattice signatures without trapdoors. In *Proc. of Eurocrypt*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.
35. V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proc. of ICALP*, volume 4052 of *LNCS*, pages 144–155. Springer, 2006.
36. V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *Proc. of TCC*, volume 4948 of *LNCS*, pages 37–54. Springer, 2008.
37. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Proc. of Eurocrypt*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
38. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. of Eurocrypt*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
39. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Proc. of Crypto*, pages 282–298, 2003.
40. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009.
41. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proc. of TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.
42. C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *Proc. of Crypto*, volume LNCS, pages 536–553. Springer, 2008.
43. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.