Simpler Efficient Group Signatures from Lattices[⋆]

Phong Q. Nguyen[1], Jiang Zhang[2], and Zhenfeng Zhang[2]

[1] INRIA, France and Tsinghua University, Institute for Advanced Study, China
[2] Trusted Computing and Information Assurance Laboratory,
State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, China
`http://www.di.ens.fr/~pnguyen`,
`jiangzhang09@gmail.com`, `zfzhang@tca.iscas.ac.cn`

Abstract. A group signature allows a group member to anonymously sign messages on behalf of the group. In the past few years, new group signatures based on lattice problems have appeared: the most efficient lattice-based constructions are due to Laguillaumie *et al.* (Asiacrypt '13) and Langlois *et al.* (PKC '14). Both have at least $O(n^2 \log^2 n \log N)$-bit group public key and $O(n \log^3 n \log N)$-bit signature, where n is the security parameter and N is the maximum number of group members. In this paper, we present a simpler lattice-based group signature, which is more efficient by a $O(\log N)$ factor in both the group public key and the signature size. We achieve this by using a new non-interactive zero-knowledge (NIZK) proof corresponding to a simple identity-encoding function. The security of our group signature can be reduced to the hardness of SIS and LWE in the random oracle model.

1 Introduction

In a group signature, each group member has a private key that is certified with its identity by the group manager. By using its private key, each group member is able to sign messages on behalf of the group without compromising its identity to the signature verifier. Group signatures provide users a nice tradeoff between authenticity and anonymity (*i.e.*, given a signature, the verifier is assured that someone in the group signed a message, but cannot determine which member of the group signed). However, such a functionality allows malicious group members to damage the whole group without being detected, *e.g.* signing some unauthorized/illegal messages. To avoid this, the group manager usually has a secret key which can be used to break anonymity.

Several real-life applications require properties of group signatures. For example, in trusted computing, a trusted platform module (TPM) usually has to attest certain statements w.r.t. the current configurations of the host device to a remote party (*i.e.* the verifier) via a signature on corresponding messages. After the attestation, the verifier is assured that some remote device that contains a TPM authorized the messages. For user privacy, the signature is often required not to reveal the identity of the TPM. In fact, a variant of group signatures (namely, direct anonymous attestation (DAA) [26,33])

[⋆] The work is supported in part by China's 973 program (No. 2013CB338003, 2013CB834205) and the National Natural Science Foundation of China (No. 61133013, 61170278, 91118006).

has been implemented in TPM 1.2 [41] and TPM 2.0 [42] by the Trusted Computing Group. Another promising application is vehicle safety communications [43], where group signatures can protect the privacy of users so that a broadcast message does not reveal the current location/speed of the vehicle. Besides, other applications of group signatures are found in anonymous communications, e-commerce systems *etc.*

Since their introduction by Chaum and van Heyst [32], group signatures have attracted much attention from the research community. Bellare, Micciancio and Warinschi (BMW) [11] formalized the security of group signatures for static groups (where the group members are fixed in the system setup phase) in two main notions, *i.e.*, *full anonymity* and *full traceability*. Informally, *full anonymity* requires that an adversary without the group manager secret key should not be able to determine the signer's identity from a signature, even if it can access an open oracle that returns the identity of any other (valid) signature. And *full traceability* implies that no collusion of group members can create a valid signature which cannot be traced back to one of them (by the group manager using the group manager secret key). Bellare *et al.* [11] also gave a theoretical construction based on the existence of trapdoor permutations. In a weak variant of the BMW model where the adversary against anonymity is not given access to the open oracle (*i.e.*, CPA-anonymity), Boneh *et al.* [16] constructed a short group signature scheme based on the Strong Diffie-Hellman (SDH) [15] and Decision Linear (DLIN) [16] assumptions in the random oracle model [13]. Besides, many papers focused on designing various group signatures based on different assumptions [9,28,20,21,39,40,1,47,46].

In recent years, lattice cryptography has attracted significant interest, due to several potential benefits: asymptotic efficiency, worst-case hardness assumptions, and security against quantum computers. A natural goal is to find lattice-based counterparts of all classical cryptographic schemes. In 2010, Gordon *et al.* [38] made the first step in constructing secure group signatures from lattices. They elegantly combined several powerful lattice-based tools [55,58,37] to build a group signature scheme where the sizes of both the group public key and signature was linear in the maximum number N of group members. Later, Camenisch el al. [29] proposed a variant of [38] with improvements both in efficiency (*i.e.*, shorter group public key) and security (*i.e.*, stronger adversary against anonymity), but the signature size of their scheme was still linear in N. Recently, two papers [44,45] have significantly decreased the signature size. By first representing the identity of group members as a bit-string [19], and then applying the "encrypt-and-prove" paradigm of [11,38], Laguillaumie *et al.* [44] constructed an efficient lattice-based group signature where both the sizes of the group public key and the signature are proportional to $\log N$ (*i.e.*, with bit-length slightly greater than $O(n^2 \log^2 n \log N)$ and $O(n \log^3 n \log N)$, respectively). Using similar identity representations together with the non-interactive zero-knowledge (NIZK) proof in [48], Langlois *et al.* [45] proposed a nice scheme without encryption, which achieves almost the same asymptotical efficiency as that of [44], and provides an additional property called verifier-local revocation [18]. Another interesting group signature is due to Benhamouda *et al.* [14], for which privacy holds under a lattice-based assumption but the security is discrete-logarithm-based, *i.e.* it is not a pure lattice-based group signature.

A current and independent work of Ling, Nguyen and Wang [49] also try to design an efficient lattice-based group signature scheme. Specifically, by first constructing a nice Stern-type [59] NIZK protocol, they propose a scheme which excels previous ones in [44,45] by a constant factor in terms of efficiency, *i.e.*, all the sizes are still proportional to $\log N$. Besides, they also show how to transform their basic scheme into the setting of ideal lattices, which can save a factor of n in the size of group public key.

1.1 Our Results

In this paper, we present a new lattice-based group signature. Compared to the best previous lattice-based schemes [38,44,45], it is both simpler and more efficient, saving a $O(\log N)$ factor in both sizes of the group public key and the signature. As in [44], we first present a simple CPA-anonymous scheme, and then extend it to a scheme with CCA-anonymity. The security of both our schemes is provably based on the hardness of the Small Integer Solutions (SIS) and Learning with Errors (LWE) problems in the random oracle model, which are both as hard as several worst-case lattice problems, such as SIVP$_\gamma$ for some polynomial factor $\gamma = poly(n)$.

In Table 1, we give a rough comparison with related lattice-based group signatures in terms of the size of the group public-key, the group user secret key and the signature. There, n denotes the security parameter, and N is the maximum number of group users. The other two parameters m and q are both polynomial in n (and N), and are usually determined by the underlying lattices used by those schemes. The integer t used in [44,45] and our scheme is a repetition parameter for obtaining NIZKs with negligible soundness error. For a security parameter n, one can set $t = \omega(\log n)$ and $m = O(n \log n)$. The choice of q might be slightly different in those schemes either for security or for functionality. For example, q is explicitly required to be larger than N in our scheme. We note that this requirement might also be satisfied in the previous three schemes for most applications. Besides, even if $N < q$ does not hold in previous schemes, the sizes of the group public-key and the signature in our scheme are still asymptotically shorter (since both N and q are polynomials in n, and $\log N = O(\log q)$ holds).

Table 1. Rough Comparison of Overheads.

Schemes	Group public-key	User secret-key	Signature	Security
GKV10 [38]	$O(nmN \log q)$	$O(nm \log q)$	$O(nmN \log q)$	CPA-anonymity
LLLS13 [44]	$O(nm \log N \log q)$	$O(nm \log q)$	$O(tm \log N \log q)$	CCA-anonymity
LLNW14 [45]	$O(nm \log N \log q)$	$O(m \log N \log q)$	$O(tm \log N \log q \log \beta)^\star$	CCA-anonymity
Our scheme	$O(nm \log q)$	$O(nm \log q)$	$O(t(m + \log N) \log q)^{\star\star}$	CCA-anonymity

* $\beta = \omega(\sqrt{n \log q \log n}) \log m$ is the integer norm bound in [45].

** Since N is always a polynomial in n (thus in m), this term is actually bounded by $O(tm \log q)$. Besides, we note that group signatures supporting opening should have a signature in bit-size at least logarithmic in N [11].

Since the schemes in [38,44] and ours follow a general "encrypt-and-prove" paradigm in [11], we also give a comparison of computational costs between the schemes in [38,44] and ours at a very high level, *i.e.*, in terms of the number of the underlying encryptions and basic NIZK proofs, in Table 2. We note that such a comparison is less interesting to the scheme in [45], since it departs from the general paradigm and does not make use of any encryption. Although all three schemes use (almost) the same encryption (namely [58]), the NIZKs are very different. Concretely, Gordon *et al.* [38] used a N-OR variant of the witness-indistinguishable (WI) proof system for the gap version of the closest vector problem in [55], while the NIZKs used in [44] and ours are derived from the more efficient protocol [51] for the ISIS problem. In Table 2, we simply compare the complexity of each algorithm of the schemes in [38,44] and ours, with respect to the number of basic operations in terms of encryptions and basic NIZKs.

Table 2. Rough Comparison of Computational Costs (The encryption and decryption of Regev's LWE-based encryption [58] are denoted by enc. and dec., respectively. The proof and verification of the corresponding basic NIZKs in [55,51] are denoted by pro. and ver., respectively)

Schemes	Sign (enc.,pro.,ver.,dec.)	Verify (enc.,pro.,ver.,dec.)	Open (enc.,pro.,ver.,dec.)
GKV10 [38]	$(N, O(N), -, -)$	$(-, -, O(N), -)$	$(-, -, -, N/2)$
LLLS13 [44]	$(1 + \log N, O(\log N), -, -)$	$(-, -, O(\log N), -)$	$(-, -, -, 1 + \log N)$
Our scheme	$(1, \leq 5, -, -)$	$(-, -, \leq 5, -)$	$(-, -, -, 1)$

However, we note that we do not provide a full comparison with all the schemes in [38,44,45], which would require at least a concrete analysis of the security reduction (running time, lattice approximation factor, success probability, *etc.*), which is usually not explicitly given in the literature.

1.2 Techniques

At a high level, the two constructions in [38,44] and our scheme use the same general paradigm as that of [11]. Roughly speaking, the group manager first generates the group public key gpk and group manager secret key $gmsk$. For a user with identity $i \in \{1, \ldots, N\}$ (recall that N is the maximum number of group members, and is fixed at the system setup), the group manager computes the user's secret key gsk_i corresponding to an encoded "public key" $H(gpk, i)$, where H is an encoding function that (uniquely) encodes the group user's identity i in gsk_i. When signing a message m, the group user proves to the verifier that he has a secret key gsk_i for some $i \in \{1, \ldots, N\}$ (*i.e.*, to prove that he is a legal member of the group). The hardness of this general paradigm usually lies in the choices of an appropriate encoding function $H(gpk, i)$ and a compatible non-interactive zero-knowledge (NIZK) for the membership relations determined by $H(gpk, i)$.

Gordon *et al.* [38] used a simple projective encoding function $H(gpk, i)$ and a NIZK extended from [55] to construct the first lattice-based group signature. Informally, the group public key gpk consists of N independent public keys of the GPV signature [37], *i.e.*, $gpk = (pk_1, \ldots, pk_N)$ where pk_j is an integer matrix over $\mathbb{Z}_q$ for some positive $q \in \mathbb{Z}$, and all $j \in \{1, \ldots, N\}$. The encoding function simply outputs the i-th element of gpk, *i.e.*, $H(gpk, i) := pk_i$. Due to the particular choice of $H(gpk, i)$, both the group public key and the signature of [38] have a size linear in N.

At Asiacrypt '13, by using an efficient encoding function inspired by Boyen's lattice-based signature [19] and a NIZK derived from [51], Laguillaumie *et al.* [44] proposed a more efficient lattice-based group signature. Roughly speaking, the group public key gpk consists of $\ell = \lfloor \log N \rfloor + 1$ independent matrices over $\mathbb{Z}_q$, *i.e.*, $gpk = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell)$. The encoding function is defined as $H(gpk, i) := \sum_{j=1}^{\ell} i_j \mathbf{A}_j$, where $(i_1, \ldots, i_l) \in \mathbb{Z}_2^\ell$ is the binary decomposition of i. We also note that Langlois *et al.* [45] constructed a lattice-based group signature with verifier-local revocation by using the same identity encoding function but a different NIZK from [48]. Both schemes [44,45] decreased the sizes of the group public key and the signature to proportional to $\log N$.

An Efficient Identity Encoding. We use a more efficient and compact way to encode the group member's identity, by building upon the encoding technique introduced by Agrawal *et al.* [2] for identity-based encryption (IBE). Let the group public key gpk consist of three matrices over $\mathbb{Z}_q^{n\times m}$ for some positive integers n, m, q, *i.e.*, $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$. We define $H(gpk, i) = \hat{\mathbf{A}}_i := (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(i)\mathbf{A}_{2,2})$ where $G(\cdot)$ is a function from $\mathbb{Z}_N$ to $\mathbb{Z}_q^{n\times n}$. Then, the secret key of user i is a short basis of the classical q-ary lattice $\mathbf{\Lambda}_i$ determined by $H(gpk, i) = \hat{\mathbf{A}}_i$, where $\mathbf{\Lambda}_i := \{\mathbf{e} \in \mathbb{Z}^m \;\; s.t. \; \hat{\mathbf{A}}_i \mathbf{e} = \mathbf{0} \mod q\}$. When signing a message, user i samples a short vector $\mathbf{e}_i$ from $\mathbf{\Lambda}_i$ (by using the short basis of $\mathbf{\Lambda}_i$) and encrypts it using Regev's encryption [58]. Then, he proves to the verifier that $\mathbf{e}_i$ is a short vector in a lattice determined by $H(gpk, i)$ for some $i \in \{1, \ldots, N\}$. But we do not know an efficient lattice-based NIZK suitable for the membership relation determined by $H(gpk, i)$.

Fortunately, since the maximum number of group members N is always bounded by a polynomial in the security parameter n, we actually do not need an encoding function as powerful as for IBE [2], where there are possibly exponentially many users. We simplify the encoding function by defining $H(gpk, i) := (\mathbf{A}_1 \| \mathbf{A}_{2,1} + i\mathbf{A}_{2,2})$. (A similar combination of matrices has been used in a different way in [4,17,36] to construct functional encryption.) Namely, the identity function $G(i) := i$ is used instead of a function $G : \mathbb{Z}_N \to \mathbb{Z}_q^{n\times n}$. For collision resistance, we require that $N < q$. Since N is usually fixed at the system setup in group signatures for static groups such as [38,44,45] and ours, one can simply set q big enough (but still a polynomial in n) to satisfy the requirement. Hereafter, we assume that $N < q$ always holds.

This encoding function provides two main benefits:

– Only three matrices are needed for the encoding function, which provides a short group public key. By comparison, there are respectively at least $O(N)$ and $O(\log N)$ matrices needed in [38] and [44].
– It gives a simple membership relation, which allows to construct an efficient NIZK proof for the relation (please see next paragraph). In [38,44], the NIZKs for relatively complex membership relations are obtained by involving many encryptions, which results in schemes with large computational costs and signature sizes.

A New Non-interactive Zero-Knowledge (NIZK). Recall that the secret key of user i is a short basis $\mathbf{T}_i$ of the q-ary lattice $\mathbf{\Lambda}_i$ determined by $\hat{\mathbf{A}}_i = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + i\mathbf{A}_{2,2})$. To sign a message, user i first samples a short vector $(\mathbf{x}_1, \mathbf{x}_2)$ by using Gentry *et al.*'s Gaussian sampling algorithm [37] such that $\mathbf{A}_1\mathbf{x}_1 + (\mathbf{A}_{2,1} + i\mathbf{A}_{2,2})\mathbf{x}_2 = \mathbf{0} \mod q$. Then, he generates an LWE encryption $\mathbf{c}$ of $\mathbf{x}_1$. The final signature σ consists of $\mathbf{c}$, $\mathbf{x}_2$, a proof π_1 that $\mathbf{c}$ encrypts $\mathbf{x}_1$ correctly, and a proof π_2 that there exists a tuple $(\mathbf{x}_1, i)$ satisfying $\mathbf{A}_1\mathbf{x}_1 + i\mathbf{A}_{2,2}\mathbf{x}_2 = -\mathbf{A}_{2,1}\mathbf{x}_2 \mod q$, namely, $\sigma = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2)$.

The nice properties of the sampling algorithm in [37,31] guarantee that the public $\mathbf{x}_2$ is statistically indistinguishable for all user $i \in \{1, \ldots, N\}$, namely, the verifier cannot determine the signer's identity i solely from $\mathbf{x}_2$, however, he can efficiently determine it from $(\mathbf{x}_1, \mathbf{x}_2)$, that's why we choose to encrypt $\mathbf{x}_1$. The proof of π_1 can be generated by using the duality of LWE and Small Integer Solutions (SIS) [52], and the NIZK proof for SIS [51] in a standard way. Thanks to our new identity encoding function $H(gpk, i)$ and the public $\mathbf{x}_2$, we manage to design a NIZK proof (*i.e.*, π_2) for the statement $\mathbf{A}_1\mathbf{x}_1 + i\mathbf{A}_{2,2}\mathbf{x}_2 = -\mathbf{A}_{2,1}\mathbf{x}_2 \mod q$ based on the hardness of SIS.

Formally, we introduce a new problem called split-SIS, which is a variant of SIS (and might be of independent interest). Given a split-SIS instance $\mathbf{A}_1, \mathbf{A}_{2,2} \in \mathbb{Z}_q^{n\times m}$, the algorithm is asked to output a triple $(\mathbf{x}_1, \mathbf{x}_2, h)$ such that $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{Z}^m$ have small norms, and $h < q = poly(n)$ is a positive integer satisfying $\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 = \mathbf{0} \mod q$. We first show that the split-SIS problem (associated with an appropriate solution space) is polynomially equivalent to the standard SIS problem. Then, we derive a family of hash functions

$$\mathcal{H} = \left\{ \begin{array}{r} f_{\mathbf{A}_1,\mathbf{A}_{2,2}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 \mod q, \mathbf{x}_2) : \\ (\mathbf{x}_1, \mathbf{x}_2, h) \in \mathbb{Z}^m \times \mathbb{Z}^m \times \mathbb{Z} \end{array} \right\}_{\mathbf{A}_1,\mathbf{A}_{2,2}\in\mathbb{Z}_q^{n\times m}}$$

from our split-SIS problem, and prove that the hash function family $\mathcal{H}$ with appropriate domain is *one-way*, *collision-resistant*, and *statistically hiding* with respect to the third input (*i.e.*, h). Combining those useful properties with the observation that $\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 = (\mathbf{A}_1\|\mathbf{A}_{2,2}\mathbf{x}_2)(\mathbf{x}_1; h) \mod q$, we manage to adapt a Σ-protocol for $\mathcal{H}$ from existing protocols for standard ISIS problems [50,51,44], which can in turn be transformed into a NIZK using the Fiat-Shamir transformation in the random oracle model. This finally helps us obtain a lattice-based group signature scheme with $O(tm \log q)$-bit signature, where the repetition parameter $t = \omega(\log n)$ is due to our NIZK as in [44,45].

In order to open a signature $\sigma = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2)$, the group manager only has to decrypt $\mathbf{c}$ to obtain $\mathbf{x}_1$, and computes an integer $h < q$ satisfying $\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 = -\mathbf{A}_{2,1}\mathbf{x}_2 \mod q$. Note that such an integer is unique if $\mathbf{A}_{2,2}\mathbf{x}_2 \neq \mathbf{0} \mod q$ for prime q. Replacing the CPA-encryption of $\mathbf{x}_1$ with a CCA one (*i.e.*, by applying the CHK transformation [30] to the IBE [2]), we obtain a CCA-anonymous group signature at a minimal price of doubling the sizes of the group public key and the signature.

1.3 On Membership Revocation

A group signature with opening allows the group manager to break the anonymity of any valid signature, however, it cannot prevent a malicious group member from using his certificate. In practice, it may be desirable to support membership revocation, *e.g.*, to revoke the certificate of a malicious group member such that he cannot sign any message in the future. Actually, membership revocation is an important and complex problem and has been extensively studied in the literature [10,18,25,27,46,47]. In [45], Langlois *et al.* constructed a lattice-based group signature with verifier-local revocation, which was the first lattice-based group signatures supporting membership revocation and achieved the same asymptotic efficiency as that of [44]. For now, we do not know how to construct a simpler and efficient group signature with membership revocation from lattices.

1.4 Roadmap

After some preliminaries, we recall several useful tools and algorithms on lattices in Section 3. In Section 4, we introduce the split-SIS problems, and construct a NIZK proof for the split-SIS problems. We finally present our CPA-anonymous group signature scheme in Section 5. The description of our CCA-anonymous group signature scheme is given in Section 6.

2 Preliminaries

2.1 Notation

The set of real numbers (integers) is denoted by $\mathbb{R}$ ($\mathbb{Z}$, resp.). By $\leftarrow_R$ we denote randomly choosing elements from some distribution (or the uniform distribution over some finite set). For a variable x following some distribution D, we denote it by $x \backsim D$. For any integer $N \in \mathbb{Z}$, we denote by $[N]$ the set of integers $\{0, 1, \ldots, N-1\}$. Vectors are in column form and denoted by bold lower-case letters (*e.g.*, $\mathbf{x}$). We view a matrix simply as the set of its column vectors and denoted by bold capital letters (*e.g.*, $\mathbf{X}$). Denote the l_2 and l_∞ norm by $\|\cdot\|$ and $\|\cdot\|_\infty$, respectively. Define the norm of a matrix $\mathbf{X}$ as the norm of its longest column (*i.e.*, $\|\mathbf{X}\| = \max_i \|\mathbf{x}_i\|$). If the columns of $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ are linearly independent, let $\widetilde{\mathbf{X}} = (\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_k)$ denote the Gram-Schmidt orthogonalization of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$ taken in that order. For $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n\times m'}$, $(\mathbf{X}\|\mathbf{Y}) \in \mathbb{R}^{n\times(m+m')}$ denotes the concatenation of the columns of $\mathbf{X}$ followed by the columns of $\mathbf{Y}$. Similarly, for $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n'\times m}$, $(\mathbf{X}; \mathbf{Y}) \in \mathbb{R}^{(n+n')\times m}$ is the concatenation of the rows of $\mathbf{X}$ followed by the rows of $\mathbf{Y}$.

Throughout this paper, we let n be the natural security parameter, so that all quantities are implicitly dependent on n. The function $\log$ denotes the natural logarithm. We will frequently use the standard notation of O, ω for classifying the growth of functions. If $f(n) = O(g(n) \cdot \log^c(n))$ for some constant c, we write $f(n) = \tilde{O}(g(n))$. By $poly(n)$ we denote some arbitrary $f(n) = O(n^c)$ for some c. We say that a function $f(n)$ is negligible if for every positive c, we have $f(n) < n^{-c}$ for sufficiently large n. We denote an arbitrary such function by $negl(n)$, and say that a probability is overwhelming if it is $1 - negl(n)$.

2.2 Group Signatures

We recall the definition and security model of group signatures. A (static) group signature scheme $\mathcal{GS}$ consists of a tuple of four Probabilistic Polynomial Time (PPT) algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$:

- $\mathsf{KeyGen}(1^n, 1^N)$: Take the security parameter n and the maximum number of group members N as inputs, output the group public key gpk, the group manager secret key $gmsk$ and a vector of users' keys $\mathbf{gsk} = (gsk_1, \ldots, gsk_N)$, where gsk_j is the j-th user's secret key for $j \in \{1, \ldots, N\}$.
- $\mathsf{Sign}(gpk, gsk_j, M)$: Take the group public key gpk, the j-th user's secret key gsk_j, and a message $M \in \{0,1\}^*$ as inputs, output a signature σ of M.
- $\mathsf{Verify}(gpk, M, \sigma)$: Take the group public key gpk, a message $M \in \{0,1\}^*$ and a string σ as inputs, return 1 if σ is a valid signature of M, else return 0.
- $\mathsf{Open}(gpk, gmsk, M, \sigma)$: Take the group public key gpk, the group manager secret key $gmsk$, a message $M \in \{0,1\}^*$, and a valid signature σ of M as inputs, output an index $j \in \{1, \ldots, N\}$ or a special symbol $\perp$ in case of opening failure.

For correctness, we require that for any $(gpk, gmsk, \mathbf{gsk}) \leftarrow \mathsf{KeyGen}(1^n, 1^N)$, any $j \in \{1, \ldots, N\}$, any message $M \in \{0,1\}^*$, and any $\sigma \leftarrow \mathsf{Sign}(gpk, gsk_j, M)$, the following conditions hold with overwhelming probability:

$$\mathsf{Verify}(gpk, M, \sigma) = 1 \; and \; \mathsf{Open}(gpk, gmsk, M, \sigma) = j$$

For group signatures, there are two security notions: anonymity and traceability [11]. The first notion, informally, says that anyone without the group manager secret key cannot determine the owner of a valid signature. The second notion says a set $\mathcal{C}$ of group members cannot collude to create a valid signature such that the Open algorithm fails to trace back to one of them. In particular, this notion implies that any non-group member cannot create a valid signature.

```
Experiment $\mathbf{Exp}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$
    $(gpk, gmsk, \mathbf{gsk}) \leftarrow \mathsf{KeyGen}(1^n, 1^N)$
    $(st, i_0, i_1, M^*) \leftarrow \mathcal{A}^{\mathsf{Open}(\cdot,\cdot)}(gpk, \mathbf{gsk})$
    $b \leftarrow_R \{0,1\}$
    $\sigma^* \leftarrow \mathsf{Sign}(gpk, gsk_{i_b}, M^*)$
    $b' \leftarrow \mathcal{A}^{\mathsf{Open}(\cdot,\cdot)}(st, \sigma^*)$
    If $b = b'$ return 1, else return 0
```

```
Experiment $\mathbf{Exp}^{\text{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$
    $(gpk, gmsk, \mathbf{gsk}) \leftarrow \mathsf{KeyGen}(1^n, 1^N)$
    $(M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\cdot,\cdot),\mathbf{Corrupt}(\cdot)}(gpk, gmsk)$
    If $\mathsf{Verify}(gpk, M^*, \sigma^*) = 0$ then return 0
    If $\mathsf{Open}(gmsk, M^*, \sigma^*) = \perp$ then return 1
    If $\exists j^* \in \{1, \ldots, N\}$ such that
        $\mathsf{Open}(gpk, gmsk, M^*, \sigma^*) = j^*$ and $j^* \notin \mathcal{C}$,
        and $(j^*, M^*)$ was not queried to $\mathsf{Sign}(\cdot,\cdot)$ by $\mathcal{A}$,
    then return 1, else return 0
```

Fig. 1. Security games for group signatures

Definition 1 (Full anonymity). *For any (static) group signature scheme $\mathcal{GS}$, we associate to an adversary $\mathcal{A}$ against the full anonymity of $\mathcal{GS}$ experiment $\mathbf{Exp}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$ in the left-side of Fig. 1, where the $\mathsf{Open}(\cdot,\cdot)$ oracle takes a valid message-signature pair (M, σ) as inputs, outputs the index of the user whose secret key is used to create σ. In the guess phase, the adversary $\mathcal{A}$ is not allowed to make an Open query with inputs (M^*, σ^*). We define the advantage of $\mathcal{A}$ in the experiment as*

$$\mathrm{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n, N) = \left| \Pr[\mathbf{Exp}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n, N) = 1] - \frac{1}{2} \right|.$$

A group signature $\mathcal{GS}$ is said to be fully anonymous if the advantage $\mathrm{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$ *is negligible in n, N for any PPT adversary $\mathcal{A}$.*

In a weak definition of anonymity (*i.e.*, CPA-anonymity), the adversary is not given access to an open oracle. In this paper, we first present a CPA-anonymous scheme, then we extend it to satisfy full/CCA anonymity.

Definition 2 (Full traceability). *For any (static) group signature scheme $\mathcal{GS}$, we associate to an adversary $\mathcal{A}$ against the full traceability of $\mathcal{GS}$ experiment* $\mathbf{Exp}^{\text{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$ *in the right-side of Fig. 1, where the* $\mathsf{Sign}(\cdot,\cdot)$ *oracle takes a user index i and a message M as inputs, returns a signature of M by using gsk_i. The* $\mathbf{Corrupt}(\cdot)$ *oracle takes a user index i as input, returns gsk_i, and $\mathcal{C}$ is a set of user indexes that $\mathcal{A}$ submitted to the* $\mathbf{Corrupt}(\cdot)$ *oracle. The advantage of $\mathcal{A}$ in the experiment is defined as*

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

A group signature $\mathcal{GS}$ is said to be fully traceable if the advantage $\mathrm{Adv}^{\text{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$ *is negligible in n, N for any PPT adversary $\mathcal{A}$.*

3 Lattices and Discrete Gaussians

An m-rank lattice $\mathbf{\Lambda} \subset \mathbb{R}^n$ is the set of all integral combinations of m linearly independent vectors $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in \mathbb{R}^{n\times m}$, *i.e.*, $\mathbf{\Lambda} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^m x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}$. The dual lattice of $\mathbf{\Lambda}$ is defined to be $\mathbf{\Lambda}^* = \left\{ \mathbf{x} \in \mathrm{span}(\Lambda) :\ \forall\, \mathbf{v} \in \mathbf{\Lambda},\ \langle \mathbf{x}, \mathbf{v}\rangle \in \mathbb{Z} \right\}$.

For $\mathbf{x} \in \mathbf{\Lambda}$, define the Gaussian function $\rho_{s,\mathbf{c}}(\mathbf{x})$ over $\mathbf{\Lambda} \subseteq \mathbb{Z}^n$ centered at $\mathbf{c} \in \mathbb{R}^n$ with parameter $s > 0$ as $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp\big(-\pi \|\mathbf{x} - \mathbf{c}\|^2/s^2 \big)$. Letting $\rho_{s,\mathbf{c}}(\mathbf{\Lambda}) = \sum_{\mathbf{x}\in\mathbf{\Lambda}} \rho_{s,\mathbf{c}}(\mathbf{x})$, define the discrete Gaussian distribution over $\mathbf{\Lambda}$ as $D_{\mathbf{\Lambda},s,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{y})}{\rho_{s,\mathbf{c}}(\mathbf{\Lambda})}$, where $\mathbf{y} \in \mathbf{\Lambda}$. The subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ (respectively) when omitted. For large enough s, almost all the elements from $D_{\mathbf{\Lambda},s,\mathbf{c}}$ are not far from $\mathbf{c}$.

Lemma 1 ([54,37]). *For any n-dimensional lattice $\mathbf{\Lambda}$ with basis $\mathbf{B} \in \mathbb{R}^{n\times n}$, vector $\mathbf{c} \in \mathbb{R}^n$, and reals $\epsilon \in (0, 1), s \geq \|\widetilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$, we have* $\Pr_{\mathbf{x}\leftarrow_R D_{\mathbf{\Lambda},s,\mathbf{c}}}[\|\mathbf{x} - \mathbf{c}\| > s\sqrt{n}] \leq \frac{1-\epsilon}{1+\epsilon} \cdot 2^{-n}$.

For any $\alpha \in \mathbb{R}^+$, integer $q \in \mathbb{Z}$, let Ψ_α be the distribution over $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. The discrete distribution $\bar{\Psi}_\alpha$ over $\mathbb{Z}_q$ is the random variable $\lfloor q \cdot X \rceil \bmod q$, where $X \leftarrow_R \Psi_\alpha$. For simplicity, we denote $\chi_\alpha := \bar{\Psi}_\alpha$.

Lemma 2 ([3]). *Let $\mathbf{e}$ be some vector in $\mathbb{Z}^m$ and let $\mathbf{y} \leftarrow_R \chi_\alpha^m$. Then the quantity $|\mathbf{e}^T\mathbf{y}|$ treated as an integer in $[0, q-1]$ satisfies $|\mathbf{e}^T\mathbf{y}| \leq \|\mathbf{e}\| q\alpha\omega(\sqrt{\log m}) + \|\mathbf{e}\|\sqrt{m}/2$ with all but negligible probability in m. In particular, if $x \leftarrow_R \chi_\alpha$ is treated as an integer in $[0, q-1]$ then $|x| \leq q\alpha\omega(\sqrt{\log m}) + 1/2$ with all but negligible probability in m.*

We also need the following three useful facts from the literature:

Lemma 3 ([37]). *Let n be a positive integer, q be a prime, and $m \geq 2n \log q$. Then for all but a $2q^{-n}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and for any $s \geq \omega(\sqrt{\log m})$, the distribution of $\mathbf{u} = \mathbf{A}\mathbf{e} \mod q$ is statistically close to uniform over $\mathbb{Z}_q^n$, where $\mathbf{e} \leftarrow_R D_{\mathbb{Z}^m,s}$.*

Lemma 4 (Generalized Leftover Hash Lemma [34]). *If $\mathcal{H} = \{H : X \to Y\}_{H \in \mathcal{H}}$ is a family of universal hash functions. Then, for any random variables $W \in X$ and I, the statistical difference between $(H, H(W), I)$ and (H, U, I) is at most $\frac{1}{2} \cdot \sqrt{2^{-\tilde{H}_\infty(W|I)}|Y|}$, where H and U are uniformly distributed over $\mathcal{H}$ and Y, respectively.*

Lemma 5 ([2]). *Let q be a prime, $m > (n+1)\log q + \omega(\log n)$ and $k = poly(n)$. Let the matrices $\mathbf{A}, \mathbf{B}, \mathbf{R}$ be uniformly and randomly chosen from $\mathbb{Z}_q^{n \times m}, \mathbb{Z}_q^{n \times k}$, and $\{-1, 1\}^{m \times k}$, respectively. Then for all vectors $\mathbf{w} \in \mathbb{Z}_q^m$, the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{R}^T\mathbf{w})$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{B}, \mathbf{R}^T\mathbf{w})$.*

3.1 Learning with Errors (LWE) and Small Integer Solutions (SIS)

Let $n \in \mathbb{Z}^+$ and $q = q(n)$ be integers, $\alpha \in \mathbb{R}^+$, χ_α be some discrete Gaussian distribution over $\mathbb{Z}_q$, and $\mathbf{s} \in \mathbb{Z}_q^n$ be some vector. Define $A_{\mathbf{s},\chi_\alpha} \subseteq \mathbb{Z}_q^n \times \mathbb{Z}_q$ as the distribution of the variable $(\mathbf{a}, \mathbf{a}^T\mathbf{s} + x)$, where $\mathbf{a} \leftarrow_R \mathbb{Z}_q^n$, $x \leftarrow_R \chi_\alpha$, and all the operations are performed in $\mathbb{Z}_q$. For m independent samples $(\mathbf{a}_1, y_1), \ldots, (\mathbf{a}_m, y_m)$ from $A_{\mathbf{s},\chi_\alpha}$, we denote it in matrix form $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$, where $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)^T$. We say that an algorithm solves LWE$_{q,\chi_\alpha}$ if, for randomly chosen $\mathbf{s} \in \mathbb{Z}_q^n$, given polynomial samples from $\mathrm{A}_{\mathbf{s},\chi_\alpha}$ it outputs $\mathbf{s}$ with overwhelming probability. The decisional variant of LWE is that, for a uniformly chosen $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, an algorithm is asked to distinguish $A_{\mathbf{s},\chi_\alpha}$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ (with only polynomial samples). For certain modulus q, the average-case decisional LWE problem is polynomially equivalent to its worst-case search version [58,56,8].

Proposition 1 ([58]). *Let $\alpha = \alpha(n) \in (0, 1)$ and let $q = q(n)$ be a prime such that $\alpha q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves LWE$_{q,\chi_\alpha}$, then there exists an efficient quantum algorithm for approximating SIVP in the l_2 norm, in the worst case, to within $\tilde{O}(n/\alpha)$ factors.*

The Small Integer Solution (SIS) problem was introduced by Ajtai [5], but its name is due to Micciancio and Regev [54], who improved Ajtai's connection between SIS and worst-case lattice problems.

Definition 3 (Small Integer Solution). *The Small Integer Solution (SIS) problem in l_2 norm is: Given an integer q, a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a real β, find a non-zero integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{0}$ (mod q) and $\|\mathbf{e}\| \leq \beta$.*

Definition 4 (Inhomogeneous Small Integer Solution). *The Inhomogeneous Small Integer Solution (ISIS) problem in l_2 norm is: Given an integer q, a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a random syndrome $\mathbf{u} \in \mathbb{Z}_q^n$, and real $\beta \in \mathbb{R}$, find an integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u}$ (mod q) and $\|\mathbf{e}\| \leq \beta$.*

The ISIS problem is an inhomogenous variant of SIS. Both problems were shown to be as hard as certain worst-case lattice problems.

Proposition 2 ([37]). *For any polynomially bounded $m, \beta = poly(n)$ and prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the average-case problems $SIS_{q,m,\beta}$ and $ISIS_{q,m,\beta}$ are as hard as approximating SIVP in the worst case to within certain $\gamma = \beta \cdot \widetilde{O}(\sqrt{n})$ factors.*

3.2 *q*-ary Lattices and Trapdoors

Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for some positive integers n, m and q. Consider the following two integer lattices:

$$\mathbf{\Lambda}_q^{\perp}(\mathbf{A}) = \left\{ \mathbf{e} \in \mathbb{Z}^m \ \ s.t. \ \mathbf{A}\mathbf{e} = 0 \mod q \right\}$$
$$\mathbf{\Lambda}_q(\mathbf{A}) = \left\{ \mathbf{y} \in \mathbb{Z}^m \ \ s.t. \ \exists \mathbf{s} \in \mathbb{Z}^n, \ \mathbf{A}^{\mathbf{T}}\mathbf{s} = \mathbf{y} \mod q \right\}$$

The two q-ary lattices defined above are dual when properly scaled, namely $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}) = q\mathbf{\Lambda}_q(\mathbf{A})^*$ and $\mathbf{\Lambda}_q(\mathbf{A}) = q\mathbf{\Lambda}_q^{\perp}(\mathbf{A})^*$. Moreover, for any $h \in \mathbb{Z}_q^*$, we have: $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}) = \mathbf{\Lambda}_q^{\perp}(h\mathbf{A})$.

In 1999, Ajtai [6] showed how to sample an essentially uniform matrix $\mathbf{A}$ together with a short basis of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$. This trapdoor generation algorithm has been significantly improved in [7,53].

Proposition 3 ([7]). *For any $\delta_0 > 0$, there is a PPT algorithm* TrapGen *that, on input a security parameter n, an odd prime $q = poly(n)$, and integer $m \geq (5 + 3\delta_0)n \log q$, outputs a statistically $(mq^{-\delta_0 n/2})$-close to uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{T_A} \subset \mathbf{\Lambda}_q^{\perp}(\mathbf{A})$ such that with overwhelming probability $\|\mathbf{T_A}\| \leq O(n \log q)$ and $\|\widetilde{\mathbf{T}}_\mathbf{A}\| \leq O(\sqrt{n \log q}) = O(\sqrt{m})$. In particular, if let $\delta_0 = \frac{1}{3}$, we can choose $m \geq \lceil 6n \log q \rceil$.*

The following proposition is implied by [31, Lem. 3.2 and Lem. 3.3], which shows that there is an efficient algorithm to extract a random basis for $(\mathbf{A}\|\mathbf{B})$ by using a short basis of $\mathbf{A}$ such that the new basis statistically hides the information of its input basis.

Proposition 4 ([31]). *There is a PPT algorithm* ExtRndBasis *which takes a matrix $\mathbf{A}' = (\mathbf{A}\|\mathbf{B}) \in \mathbb{Z}_q^{n \times (m+m')}$, a basis $\mathbf{T_A} \in \mathbb{Z}_q^{m \times m}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, an arbitrary matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$, and a real $s \geq \|\widetilde{\mathbf{T}}_\mathbf{A}\| \cdot \omega(\sqrt{\log m})$ as inputs, outputs a random basis $\mathbf{T}_{\mathbf{A}'}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}')$ satisfying $\|\mathbf{T}_{\mathbf{A}'}\| \leq s(m + m')$ and $\|\widetilde{\mathbf{T}}_{\mathbf{A}'}\| \leq s\sqrt{m + m'}$.*

Equipped with the above proposition, and the proof technique of [2, Th. 4], we obtain the following useful proposition:

Proposition 5. *Let $q > 2, m > n$, there is a PPT algorithm* `ExtBasisRight` *which takes matrix $\mathbf{A}' = (\mathbf{C}\|\mathbf{A}\|\mathbf{AR} + \mathbf{B}) \in \mathbb{Z}_q^{n \times (2m+m')}$, a uniformly and randomly chosen $\mathbf{R} \in \{-1, 1\}^{m \times m}$, a basis $\mathbf{T_B}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{B})$, arbitrary $\mathbf{C} \in \mathbb{Z}_q^{n \times m'}$ and a Gaussian parameter $s > \|\widetilde{\mathbf{T}}_\mathbf{B}\| \cdot \sqrt{m}\omega(\log m)$, outputs a basis $\mathbf{T}_{\mathbf{A}'}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}')$ satisfying $\|\mathbf{T}_{\mathbf{A}'}\| \leq s(2m + m')$ and $\|\widetilde{\mathbf{T}}_{\mathbf{A}'}\| \leq s\sqrt{2m + m'}$.*

Proof. As shown in the proof of [2, Th. 4], we can use $\mathbf{T_B}$ to efficiently sample a basis $\mathbf{T}_{\hat{\mathbf{A}}}$ for the matrix $\hat{\mathbf{A}} = (\mathbf{A}\|\mathbf{AR}+\mathbf{B})$ satisfying $\|\widetilde{\mathbf{T}}_{\hat{\mathbf{A}}}\| \leq \|\widetilde{\mathbf{T}}_{\mathbf{B}}\| \cdot \sqrt{m}\omega(\sqrt{\log m})$, then we can apply Proposition 4 to obtain a basis $\mathbf{T}_{\mathbf{A}'}$ for $\mathbf{A}' = (\mathbf{C}\|\hat{\mathbf{A}})$ satisfying $\|\widetilde{\mathbf{T}}_{\mathbf{A}'}\| \leq s\sqrt{2m+m'}$. Besides, by the property of the ExtRndBasis algorithm, the claim still holds no matter how the (columns of) matrix $\mathbf{C}$ appears in $\mathbf{A}'$. □

The following SuperSamp algorithm allows us to sample a random matrix $\mathbf{B}$ together with a short basis such that the columns of $\mathbf{B}$ lie in a prescribed affine subspace of $\mathbb{Z}_q^n$.

Proposition 6 ([44]). *Let $q > 2, m > \lceil 6n\log q + n\rceil$, there is a PPT algorithm* SuperSamp *which takes matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{C} \in \mathbb{Z}_q^{n\times n}$ as inputs, and outputs an almost uniform matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ such that $\mathbf{AB}^T = \mathbf{C}$, and a basis $\mathbf{T_B}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{B})$ satisfying $\|\mathbf{T}_B\| \leq m^{1.5}\cdot\omega(\sqrt{\log m})$ and $\|\widetilde{\mathbf{T}}_B\| \leq m\cdot\omega(\sqrt{\log m})$.*

Given a basis of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, there is an efficient algorithm to solve the (I)SIS problem as follows.

Proposition 7 ([37]). *There is a PPT algorithm* SamplePre *that, given a basis $\mathbf{T_A}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, a real $s \geq \|\widetilde{\mathbf{T}}_{\mathbf{A}}\|\cdot\omega(\sqrt{\log m})$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, outputs a vector $\mathbf{e} \backsim D_{\mathbb{Z}^m,s}$ satisfying $\mathbf{Ae} = \mathbf{u}$.*

3.3 Non-interactive Zero-Knowledge Proofs of Knowledge

In 2013, Laguillaumie *et al.* [44] adapted the protocol of [50,51] to obtain a zero-knowledge proof of knowledge for the ISIS problem in the random oracle model. Concretely, there is a non-interactive zero-knowledge proof of knowledge (NIZKPoK) for the ISIS relations

$$R_{\text{ISIS}} = \{(\mathbf{A},\mathbf{y},\beta;\mathbf{x}) \in \mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^n\times\mathbb{R}\times\mathbb{Z}^m : \mathbf{Ax}=\mathbf{y} \text{ and } \|\mathbf{x}\| \leq \beta\}.$$

In particular, there is a knowledge extractor which, given two valid proofs with the same commitment message but two different challenges, outputs a witness $\mathbf{x}'$ satisfying $\|\mathbf{x}'\| \leq O(\beta m^2)$ and $\mathbf{Ax}' = \mathbf{y}$. By using the duality between LWE and ISIS, there exists an NIZKPoK for the LWE relation:

$$R_{\text{LWE}} = \{(\mathbf{A},\mathbf{b},\alpha;\mathbf{s}) \in \mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^m\times\mathbb{R}\times\mathbb{Z}_q^n : \|\mathbf{b}-\mathbf{A}^T\mathbf{s}\| \leq \alpha q\sqrt{m}\}.$$

Actually, as noted in [52], given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ such that the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$ (this holds with overwhelming probability for a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$), one can compute a matrix $\mathbf{G} \in \mathbb{Z}_q^{(m-n)\times m}$ such that 1) the columns of $\mathbf{G}$ generate $\mathbb{Z}_q^{m-n}$; 2) $\mathbf{GA}^T = \mathbf{0}$. Thus, to prove $(\mathbf{A},\mathbf{b},\alpha;\mathbf{s}) \in R_{\text{LWE}}$, one can instead prove the existence of $\mathbf{e}$ such that $\|\mathbf{e}\| \leq \alpha q\sqrt{m}$ and $\mathbf{Ge} = \mathbf{Gb}$. In particular, in the construction of our group signature we need to prove that for given $(\mathbf{A},\mathbf{b}) \in \mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^m$, there exist short vectors $(\mathbf{e},\mathbf{x})$ such that $\|\mathbf{e}\| \leq \alpha q\sqrt{m}$, $\|\mathbf{x}\| \leq \beta$ and $\mathbf{b} = \mathbf{A}^T\mathbf{s}+p\mathbf{e}+\mathbf{x}$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, where $p \geq (\alpha q\sqrt{m}+\beta)m^2$. Similarly, this can also be achieved by proving the existence of short vectors $\mathbf{e}$ and $\mathbf{x}$ such that $p\mathbf{Ge}+\mathbf{Gx} =$

$\mathbf{Gb}$ using the NIZKPoK for ISIS relations. Formally, denoting $\gamma = \max(\alpha q\sqrt{m}, \beta)$, there exists an NIZKPoK for the extended-LWE (eLWE) relations

$$R_{\text{eLWE}} = \{(\mathbf{A}, \mathbf{b}, \gamma; \mathbf{s}, \mathbf{e}, \mathbf{x}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m \times \mathbb{R} \times \mathbb{Z}_q^n \times \mathbb{Z}^{2m} : \\ \mathbf{b} = \mathbf{A}^T\mathbf{s} + p\mathbf{e} + \mathbf{x} \text{ and } \|\mathbf{e}\| \leq \gamma \text{ and } \|\mathbf{x}\| \leq \gamma\}.$$

4 Split-SIS Problems

Given uniformly random matrices $(\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m}$, integer $N = N(n)$ and $\beta = \beta(n)$, an algorithm solving the split-SIS$_{q,m,\beta,N}$ problem is asked to output a tuple $(\mathbf{x} = (\mathbf{x}_1; \mathbf{x}_2), h) \in \mathbb{Z}^{2m} \times \mathbb{Z}$ such that

- $\mathbf{x}_1 \neq 0$ or $h\mathbf{x}_2 \neq 0$
- $\|\mathbf{x}\| \leq \beta$, $h \in [N]$, and $\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2 = 0$.

Recall that the standard SIS$_{q,m',\beta}$ problem asks an algorithm to find a root of the hash function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} = \mathbf{0} \mod q$ for a uniformly chosen matrix $\mathbf{A}$ and a "narrow" domain $\hat{D}_{m',\beta} := \{\mathbf{x} \in \mathbb{Z}^{m'} : \|\mathbf{x}\| \leq \beta\}$. While for the split-SIS$_{q,m,\beta,N}$ problem, the algorithm is allowed to "modify" the function by defining $f_{\mathbf{A}'}(\mathbf{x}') = \mathbf{A}'\mathbf{x}'$ for $\mathbf{A}' = (\mathbf{A}_1\|\mathbf{A}_2\mathbf{x}_2)$ with arbitrarily $\mathbf{x}_2 \in \hat{D}_{m,\beta}$, and outputs a root $\mathbf{x}' = (\mathbf{x}_1, h) \in \hat{D}_{m,\beta} \times [N]$. Intuitively, the split-SIS$_{q,m,\beta,N}$ problem is not harder than SIS$_{q,2m,\beta}$ problem. Since if $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ is a solution of the SIS$_{q,2m,\beta}$ instance $\mathbf{A} = (\mathbf{A}_1\|\mathbf{A}_2)$, $(\mathbf{x}, 1)$ is a solution of the split-SIS$_{q,m,\beta,N}$ instance $(\mathbf{A}_1, \mathbf{A}_2)$ with $N \geq 1$.

However, for prime $q = q(n)$, and $N = N(n) < q$ of a polynomial in n, we show in the following theorem that the split-SIS$_{q,m,\beta,N}$ problem is at least as hard as SIS$_{q,2m,\beta}$. Thus, the average-case hardness of the split-SIS problem is based on the worst-case hardness of SIVP by Proposition 2.

Theorem 1 (Hardness of Split-SIS Problems). *For any polynomial $m = m(n), \beta = \beta(n), N = N(n)$, and any prime $q \geq \beta \cdot \omega(\sqrt{n \log n}) > N$, the split-SIS$_{q,m,\beta,N}$ problem is polynomially equivalent to SIS$_{q,2m,\beta}$ problem. In particular, the average-case split-SIS$_{q,m,\beta,N}$ is as hard as approximating the SIVP problem in the worst case to within certain $\gamma = \beta \cdot \widetilde{O}(\sqrt{n})$ factors.*

Proof. The direction from split-SIS$_{q,m,\beta,N}$ to SIS$_{q,2m,\beta}$ is obvious. We now prove the other direction. Assume that there is an algorithm $\mathcal{A}$ that solves split-SIS$_{q,m,\beta,N}$ with probability ϵ, we now construct an algorithm $\mathcal{B}$ that solves SIS$_{q,2m,\beta}$ with probability at least ϵ/N (recall that N is a polynomial in n). Formally, given a SIS$_{q,2m,\beta}$ instance $\hat{\mathbf{A}} = (\hat{\mathbf{A}}_1\|\hat{\mathbf{A}}_2) \in \mathbb{Z}_q^{n\times 2m}$, $\mathcal{B}$ randomly chooses an integer $h^* \leftarrow_R [N]$. If $h^* = 0$, $\mathcal{B}$ sets $\mathbf{A} = \hat{\mathbf{A}}$. Otherwise, $\mathcal{B}$ sets $\mathbf{A} = (h^*\hat{\mathbf{A}}_1\|\hat{\mathbf{A}}_2)$. Since q is a prime and $N < q$ (*i.e.*, $h^* \neq 0$ is invertible in $\mathbb{Z}_q$), we have that $\mathbf{A}$ is uniformly distributed over $\mathbb{Z}_q^{n\times 2m}$. Then, $\mathcal{B}$ gives $\mathbf{A} = (\mathbf{A}_1\|\mathbf{A}_2)$ to $\mathcal{A}$, and obtains a solution $(\mathbf{x} = (\mathbf{x}_1; \mathbf{x}_2), h) \in \mathbb{Z}^{2m} \times [N]$ satisfying $\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2 = \mathbf{0}$. If $h^* \neq h$, $\mathcal{B}$ aborts. (Since h^* is randomly chosen from $[N]$, the probability $\Pr[h^* = h]$ is at least $1/N$.) Otherwise, $\mathcal{B}$ returns $\mathbf{y} = (\mathbf{x}_1; \mathbf{0})$ if $h^* = 0$, else returns $\mathbf{y} = \mathbf{x}$. The first claim follows from the fact that $\mathbf{y} \neq 0$, $\|\mathbf{y}\| \leq \beta$ and $\hat{\mathbf{A}}\mathbf{y} = 0$. Combining this with Proposition 2, the second claim follows. □

4.1 A Family of Hash functions from Split-SIS Problems

We define a new family of hash functions based on the split-SIS problem, which plays a key role in reducing the sizes of the group public key and the signature in our construction. Formally, for integers n, m, prime q, and polynomial $\beta = \beta(n) \geq \omega(\sqrt{\log m})$, $N = N(n) < q$, we define $D_{m,\beta} = \{\mathbf{x} \leftarrow_R D_{\mathbb{Z}^m,\beta} : \|\mathbf{x}\| \leq \beta\sqrt{m}\}$, and a hash function family $\mathcal{H}_{n,m,q,\beta,N} = \{f_{\mathbf{A}} : D_{m,\beta,N} \to \mathbb{Z}_q^n \times D_{m,\beta}\}_{\mathbf{A}\in\mathbb{Z}_q^{n\times 2m}}$, where $D_{m,\beta,N} := D_{m,\beta} \times D_{m,\beta} \times [N]$. For index $\mathbf{A} = (\mathbf{A}_1\|\mathbf{A}_2) \in \mathbb{Z}_q^{n\times 2m}$, and input $(\mathbf{x}_1, \mathbf{x}_2, h) \in D_{m,\beta,N}$, the hash value $f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{x}_2, h) := (\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2, \mathbf{x}_2) \in \mathbb{Z}_q^n \times D_{m,\beta}$. In the following, we show three properties of $\mathcal{H}_{n,m,q,\beta,N}$, which are useful to construct zero-knowledge proofs for the function in $\mathcal{H}_{n,m,q,\beta,N}$.

Theorem 2 (One-Wayness). *For parameters $m > 2n\log q, \beta = \beta(n) > 2\cdot\omega(\sqrt{\log m})$, prime $q = q(n)$, and polynomial $N = N(n) < q$, if the split-SIS$_{q,m,\sqrt{5m}\beta,N}$ problem is hard, then the family of hash functions $\mathcal{H}_{n,m,q,\beta,N}$ is one-way.*

Proof. Assume that there is an algorithm $\mathcal{A}$ that breaks the one-wayness of $\mathcal{H}_{n,m,q,\beta,N}$, we construct an algorithm $\mathcal{B}$ that solves the split-SIS$_{q,m,\sqrt{5m}\beta,N}$ problem. Actually, given a split-SIS$_{q,m,\sqrt{5m}\beta,N}$ instance $\mathbf{A} = (\mathbf{A}_1\|\mathbf{A}_2) \in \mathbb{Z}_q^{n\times 2m}$, $\mathcal{B}$ randomly chooses $(\mathbf{x}_1, \mathbf{x}_2) \in D_{\mathbb{Z}^m,\beta} \times D_{\mathbb{Z}^m,\beta}$ and $h \leftarrow_R [N]$, and computes $\mathbf{y} = f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2, \mathbf{x}_2)$. Then, it gives $(\mathbf{A}, \mathbf{y})$ to $\mathcal{A}$, and obtains $(\mathbf{x}'_1, \mathbf{x}'_2, h')$ satisfying $(\mathbf{A}_1\mathbf{x}'_1 + h'\mathbf{A}_2\mathbf{x}'_2, \mathbf{x}'_2) = \mathbf{y}$. Finally, if $h \geq h'$, $\mathcal{B}$ outputs $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{h}) = (\mathbf{x}_1 - \mathbf{x}'_1, \mathbf{x}_2, h - h')$. Else, $\mathcal{B}$ outputs $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{h}) = (\mathbf{x}'_1 - \mathbf{x}_1, \mathbf{x}_2, h' - h)$.

It is easy to check that $\mathbf{A}_1\hat{\mathbf{x}}_1 + \hat{h}\mathbf{A}_2\hat{\mathbf{x}}_2 = \mathbf{0} \mod q$, $\hat{h} \in [N]$, and $\|(\hat{\mathbf{x}}_1; \hat{\mathbf{x}}_2)\| \leq \sqrt{5m}\beta$ with overwhelming probability by the standard tail inequality of the Gaussian distribution $D_{\mathbb{Z}^m,\beta}$. We finish this proof by showing that $\Pr[\hat{\mathbf{x}}_1 = \mathbf{0}]$ is negligible in n. Note that $\mathcal{A}$ can only obtain the information about $\mathbf{x}_1$ from $\mathbf{A}_1\mathbf{x}_1$. By [37, Lem. 5.2], this only leaks the distribution $\mathbf{t} + D_{\Lambda_q^\perp(\mathbf{A}_1),\beta,-\mathbf{t}}$ for any $\mathbf{t}$ satisfying $\mathbf{A}_1\mathbf{t} = \mathbf{A}_1\mathbf{x}_1$. Namely, $\mathbf{x}_1$ should be uniformly distributed over $\mathbf{t} + D_{\Lambda_q^\perp(\mathbf{A}_1),\beta,-\mathbf{t}}$ from the view of $\mathcal{A}$. Combining this with [57, Lem. 2.16], we have $\Pr[\mathbf{x}_1 = \mathbf{x}'_1]$ is negligible in n. In other words, we have $\Pr[\hat{\mathbf{x}} \neq \mathbf{0}] = 1 - negl(n)$, which completes the proof. □

Since $f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{0}, h) = f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{0}, 0)$ holds for all $h \in [N]$, the function $\mathcal{H}_{n,m,q,\beta,N}$ with domain $D_{m,\beta,N} := D_{m,\beta} \times D_{m,\beta} \times [N]$ are not collision-resistant. However, if we slightly restrict the domain of $\mathcal{H}_{n,m,q,\beta,N}$ to exclude the above trivial case, we can prove that the family of $\mathcal{H}_{n,m,q,\beta,N}$ is collision-resistant. Formally, we slightly restrict the domain of $\mathcal{H}_{n,m,q,\beta,N}$ to be $D'_{m,\beta,N} = \{(\mathbf{x}_1, \mathbf{x}_2, h) \in D_{m,\beta,N} : \mathbf{x}_2 \neq \mathbf{0}\}$.

Theorem 3 (Collision-Resistance). *For parameter $m = m(n), \beta = \beta(n)$, prime $q = q(n)$, and polynomial $N = N(n) < q$, if the split-SIS$_{q,m,\sqrt{5m}\beta,N}$ problem is hard, then the family of hash functions $\mathcal{H}_{n,m,q,\beta,N}$ with domain $D'_{m,\beta,N}$ is collision-resistant.*

Proof. Assume there is a PPT algorithm $\mathcal{A}$ that can find collisions of $\mathcal{H}_{n,m,q,\beta,N}$ with non-negligible probability ϵ, we construct an algorithm $\mathcal{B}$ solving split-SIS$_{q,m,\sqrt{5m}\beta,N}$ with the same probability. Concretely, after obtaining a split-SIS$_{q,m,\sqrt{5m}\beta,N}$ instance $\mathbf{A} = (\mathbf{A}_1\|\mathbf{A}_2)$, $\mathcal{B}$ directly gives $\mathbf{A}$ to $\mathcal{A}$, and obtains a pair of collisions $(\mathbf{x}_1, \mathbf{x}_2, h) \in D'_{m,\beta,N}$ and $(\mathbf{x}'_1, \mathbf{x}'_2, h') \in D'_{m,\beta,N}$ satisfying $(\mathbf{x}_1, \mathbf{x}_2, h) \neq (\mathbf{x}'_1, \mathbf{x}'_2, h')$ and $f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{x}_2, h) = f_{\mathbf{A}}(\mathbf{x}'_1, \mathbf{x}'_2, h')$. Note that in this case, we must have $\mathbf{x}_2 = \mathbf{x}'_2 \neq \mathbf{0}$. If

$h \geq h'$, $\mathcal{B}$ returns $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{h}) = (\mathbf{x}_1 - \mathbf{x}'_1, \mathbf{x}_2, h - h')$, else it returns $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{h}) = (\mathbf{x}'_1 - \mathbf{x}_1, \mathbf{x}_2, h' - h)$. By the assumption that $(\mathbf{x}_1, \mathbf{x}_2, h) \neq (\mathbf{x}'_1, \mathbf{x}'_2, h')$, the inequality $(\hat{\mathbf{x}}_1, \hat{h}) \neq \mathbf{0}$ holds in both cases, *i.e.*, we always have $\hat{\mathbf{x}}_1 \neq \mathbf{0}$ or $\hat{h}\hat{\mathbf{x}}_2 \neq \mathbf{0}$. The claim follows from the fact that $\|(\hat{\mathbf{x}}_1; \hat{\mathbf{x}}_2)\| \leq \sqrt{5m}\beta$ and $\hat{h} \in [N]$. □

Finally, we show that the family of hash functions $\mathcal{H}_{n,m,q,\beta,N}$ statistically hides its third input.

Theorem 4. *Let parameter $m > 2n \log q, \beta = \beta(n) > \omega(\sqrt{\log m})$, prime $q = q(n)$, and polynomial $N = N(n)$. Then, for a randomly chosen $\mathbf{A} = (\mathbf{A}_1 \| \mathbf{A}_2) \in \mathbb{Z}_q^{n \times 2m}$, and arbitrarily $\mathbf{x}_2$ with norm $\|\mathbf{x}_2\| \leq \beta\sqrt{m}$, the statistical distance between the following two distributions:*

$$\{(\mathbf{A}, f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{x}_2, h), h) : \mathbf{x}_1 \leftarrow_R D_{m,\beta}, h \leftarrow_R [N]\}$$

and

$$\{(\mathbf{A}, (\mathbf{u}, \mathbf{x}_2), h) : \mathbf{u} \leftarrow_R \mathbb{Z}_q^n, h \leftarrow_R [N]\}$$

is negligible in n.

Proof. Since the second output of $f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{x}_2, h)$ (*i.e.*, $\mathbf{x}_2$) is independent from the choices of h, we only have to show that, for arbitrarily $\mathbf{x}_2$ and h, the distribution $\{\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2 : \mathbf{x}_1 \leftarrow_R D_{m,\beta}\}$ is statistically close to uniform over $\mathbb{Z}_q^n$. Actually, using the fact that $\beta \geq \omega(\sqrt{\log m})$ together with Lemma 3, we have that the distribution of $\mathbf{A}_1\mathbf{x}_1$ is statistically close to uniform over $\mathbb{Z}_q^n$ when $\mathbf{x}_1 \leftarrow_R D_{\mathbb{Z}^m,\beta}$. The claim of this theorem follows from the fact that the statistical distance between $D_{\mathbb{Z}^m,\beta}$ and $D_{m,\beta}$ is negligible, and that the distribution $\{\mathbf{u} + h\mathbf{A}_2\mathbf{x}_2 : \mathbf{u} \leftarrow_R \mathbb{Z}_q^n\}$ is exactly the uniform distribution over $\mathbb{Z}_q^n$ for arbitrary $\mathbf{x}_2 \in D_{m,\beta}, h \in [N]$. □

4.2 Zero-Knowledge Proof of Knowledge for the Hash Functions

In this subsection, we present a proof of knowledge protocol for the family of hash functions $\mathcal{H}_{n,m,q,\beta,N}$. Concretely, given a matrix $\mathbf{A} = (\mathbf{A}_1 \| \mathbf{A}_2)$, a vector $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ with $0 < \|\mathbf{y}_2\| \leq \beta\sqrt{m}$, the prover can generate a proof of knowledge of $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, h) \in \mathbb{Z}^{2m+1}$ satisfying $\|\mathbf{x}_1\| \leq \beta\sqrt{m}$, $h \in [N]$ and $f_{\mathbf{A}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2, \mathbf{x}_2) = \mathbf{y}$. Since $\mathbf{x}_2$ must be equal to $\mathbf{y}_2$, the protocol is actually a proof of knowledge for the relation

$$R_{\text{split-SIS}} = \{(\mathbf{A}, \mathbf{y}, \beta, N; \mathbf{x}_1, h) \in \mathbb{Z}_q^{n \times 2m} \times (\mathbb{Z}_q^n \times \mathbb{Z}^m) \times \mathbb{R} \times \mathbb{Z} \times \mathbb{Z}^m \times \mathbb{Z} : \\ \mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{y}_2 = \mathbf{y}_1, \|\mathbf{x}_1\| \leq \beta\sqrt{m} \text{ and } h \in [N]\}.$$

Intuitively, we can adapt a variant of the protocols for ISIS relations in [50,51,44] for our purpose, since one can rewrite $\mathbf{y}_1 = \mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{y}_2 = (\mathbf{A}_1 \| \mathbf{A}_2\mathbf{y}_2)(\mathbf{x}_1; h)$. However, this may not work when $N \gg \beta$. Since the basic idea of [50,51,44] is to use randomness from a "large width" distribution (compared to the distribution of the witness) to hide the distribution of the witness, the width of the randomness distribution should be sufficiently larger than N in our case, which might lead to a proof without soundness guarantee.

Fortunately, we can borrow the "bit-decomposition" technique from [23,22,24] to deal with large N. The idea is to decompose $h \in [N]$ into a vector of small elements, and then prove the existence of such a vector for h. Formally, for any $h \in [N]$, we compute the representation of h in base $\bar{\beta} = \lfloor \beta \rfloor$, namely, a ℓ-dimension vector $\mathbf{v}_h = (v_0, \ldots, v_{\ell-1}) \in \mathbb{Z}^\ell$ such that $0 \le v_i \le \beta - 1$ and $h = \sum_{i=0}^{\ell-1} v_i \bar{\beta}^i$, where $\ell = \lceil \log_{\bar{\beta}} N \rceil$. Denote $\mathbf{b} = \mathbf{A}_2\mathbf{y}_2$, compute $\mathbf{D} = (\mathbf{b}, \bar{\beta}\mathbf{b}, \ldots, \bar{\beta}^{\ell-1}\mathbf{b}) \in \mathbb{Z}_q^{n\times\ell}$. It is easy to check that for any vector $\mathbf{e} \in \mathbb{Z}^\ell$, there exists a $h' \in \mathbb{Z}_q$ such that $\mathbf{De} = h'\mathbf{b} \mod q$. ($h' \in \mathbb{Z}_q$ is unique if $\mathbf{b} \neq \mathbf{0}$.) In particular, we have that $\mathbf{y}_1 = \hat{\mathbf{A}}\hat{\mathbf{x}}$, where $\hat{\mathbf{A}} = (\mathbf{A}_1 \| \mathbf{D}) \in \mathbb{Z}_q^{n\times(m+\ell)}$, $\hat{\mathbf{x}} = (\mathbf{x}_1; \mathbf{v}_h) \in \mathbb{Z}^{m+\ell}$ and $\|\hat{\mathbf{x}}\| \le \beta\sqrt{m+\ell}$. Since $\bar{\beta} > 2$ and N is a polynomial in n, we have $\ell \ll m$ and $\|\hat{\mathbf{x}}\| < \eta = \beta\sqrt{2m}$.

We first present a Σ-protocol for the function family $\mathcal{H}_{n,m,q,\beta,N}$, which repeats a basic protocol with single-bit challenge $t = \omega(\log n)$ times in parallel. As in [51,44], the basic protocol makes use of the rejection sampling technique to achieve zero-knowledge. Formally, let $\gamma = \eta \cdot m^{1.5}$, denote $\zeta(\mathbf{z}, \mathbf{y}) = 1 - \min(\frac{D_{\mathbb{Z}^{m+\ell},\gamma}(\mathbf{z})}{M_l \cdot D_{\mathbb{Z}^{m+\ell},\mathbf{y},\gamma}(\mathbf{z})}, 1)$, where $\mathbf{y}, \mathbf{z} \in \mathbb{Z}^{m+\ell}$, and the constant $M_l \le 1 + O(\frac{1}{m})$ is set according to Lemma 4.5 in [51], the protocol is depicted in Fig 2.

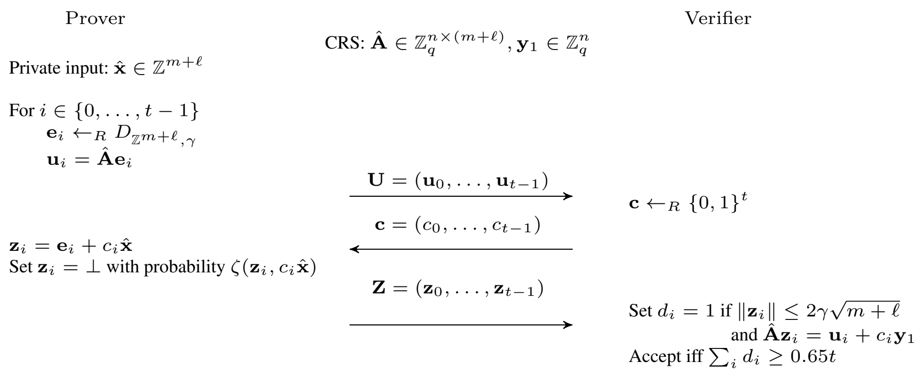
Prover
Verifier
CRS: $\hat{\mathbf{A}} \in \mathbb{Z}_q^{n\times(m+\ell)}, \mathbf{y}_1 \in \mathbb{Z}_q^n$
Private input: $\hat{\mathbf{x}} \in \mathbb{Z}^{m+\ell}$
For $i \in \{0, \ldots, t-1\}$
$\mathbf{e}_i \leftarrow_R D_{\mathbb{Z}^{m+\ell},\gamma}$
$\mathbf{u}_i = \hat{\mathbf{A}}\mathbf{e}_i$
$\mathbf{U} = (\mathbf{u}_0, \ldots, \mathbf{u}_{t-1})$
$\mathbf{c} \leftarrow_R \{0,1\}^t$
$\mathbf{c} = (c_0, \ldots, c_{t-1})$
$\mathbf{z}_i = \mathbf{e}_i + c_i\hat{\mathbf{x}}$
Set $\mathbf{z}_i = \perp$ with probability $\zeta(\mathbf{z}_i, c_i\hat{\mathbf{x}})$
$\mathbf{Z} = (\mathbf{z}_0, \ldots, \mathbf{z}_{t-1})$
Set $d_i = 1$ if $\|\mathbf{z}_i\| \le 2\gamma\sqrt{m+\ell}$
and $\hat{\mathbf{A}}\mathbf{z}_i = \mathbf{u}_i + c_i\mathbf{y}_1$
Accept iff $\sum_i d_i \ge 0.65t$

Fig. 2. Σ-protocol for $R_{\text{split-SIS}}$

By [51, Th. 4.6] we have $\Pr[\mathbf{z}_i \neq \perp] \approx \frac{1}{M_l} = 1 - O(\frac{1}{m})$ for each $i \in \{0, \ldots, t-1\}$. In addition, $\Pr[\|\mathbf{z}_i\| \le 2\gamma\sqrt{m+\ell} \mid \mathbf{z}_i \neq \perp] = 1 - negl(m)$ by [51, Lem. 4.4]. A simple calculation shows that the completeness error of the protocol is at most $2^{-\Omega(t)}$ (when m is sufficiently large, *e.g.*, $m > 100$). Besides, the protocol has the property of special Honest-Verifier Zero Knowledge (HVZK). Namely, given a challenge c_i, there exists a simulator $\mathcal{S}$ that outputs a distribution $(\mathbf{u}_i, c_i, \mathbf{z}_i)$ statistically close to the real transcript distribution. Concretely, $\mathcal{S}$ first chooses $\mathbf{z}_i \leftarrow_R D_{\mathbb{Z}^{m+\ell},\gamma}$, and computes $\mathbf{u}_i = \hat{\mathbf{A}}\mathbf{z}_i - c_i\mathbf{y}_1 \mod q$. Then, it sets $\mathbf{z}_i = \perp$ with probability $1 - \frac{1}{M_l}$, and outputs $(\mathbf{u}_i, c_i, \mathbf{z}_i)$. By Theorem 4, the term $\hat{\mathbf{A}}\mathbf{z}_i(\mod q)$ is statistically close to uniform over $\mathbb{Z}_q^n$, thus the distribution of $\mathbf{u}_i$ is statistically close to that in the real proof. Moreover, by [51, Th. 4.6], the distribution of $\mathbf{z}_i$ is also statistically close to that in the real transcripts.

Finally, since the binary challenges (*i.e.*, $\mathbf{c}$) are used, the above protocol has the property of special soundness. Actually, given two transcripts $(\mathbf{U}, \mathbf{c}, \mathbf{Z})$ and $(\mathbf{U}, \mathbf{c}', \mathbf{Z}')$ with distinct challenges $\mathbf{c} \neq \mathbf{c}'$, one can extract a "weak" witness $\mathbf{x}' = \mathbf{z}_i - \mathbf{z}'_i$ for some i satisfying $\hat{\mathbf{A}}\mathbf{x}' = \mathbf{y}_1$ and $\|\mathbf{x}'\| \leq 4\gamma\sqrt{2m}$.

Applying the "Fiat-Shamir Heuristic" [35] in a standard way, one can obtain an NIZKPoK by computing $\mathbf{c} = H(\rho, \mathbf{U})$, where $H : \{0,1\}^* \to \{0,1\}^t$ is modeled as a random oracle, and ρ represents all the other auxiliary inputs, *e.g.*, a specified message M to be signed. Finally, due to the nice property of Σ-protocol, one can easily combine the protocol to prove EQ-relation, OR-relation, and AND-relation, we omit the details.

5 A Simple and Efficient Group Signature from Lattices

In this section, we present our CPA-anonymous lattice-based group signature, which can be easily extended to support CCA anonymity by replacing the underlying encryption with a CCA one.

5.1 Our Construction

Assume that the security parameter is n, and δ is a real such that $n^{1+\delta} > \lceil (n+1)\log q + n \rceil$, all other parameters $m, s, \alpha, \beta, \eta, p, q$ are determined as follows:

$$\begin{aligned}
&m = 6n^{1+\delta}\\
&s = m \cdot \omega(\log m)\\
&\beta = s\sqrt{2m} \cdot \omega(\sqrt{\log 2m}) = m^{1.5} \cdot \omega(\log^{1.5} m)\\
&p = m^{2.5}\beta = m^4 \cdot \omega(\log^{1.5} m)\\
&q = m^2 \cdot \max(pm^{2.5} \cdot \omega(\log m), 4N) = m^{2.5} \max(m^6 \cdot \omega(\log^{2.5} m), 4N)\\
&\alpha = 2\sqrt{m}/q\\
&\eta = \max(\beta, \alpha q)\sqrt{m} = m^2 \cdot \omega(\log^{1.5} m)
\end{aligned} \tag{1}$$

Now, we present our group signature $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$:

- $\mathsf{KeyGen}(1^n, 1^N)$: Take the security parameter n and the maximum number N of group members as inputs, set an integer $m \in \mathbb{Z}$, primes $p, q \in \mathbb{Z}$, and $s, \alpha, \beta, \eta \in \mathbb{R}$ as above, and choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$ (modeled as random oracle) for the NIZKPoK proof, where $t = \omega(\log n)$. Then, the algorithm proceeds as follows:
  1. Compute $(\mathbf{A}_1, \mathbf{T}_{\mathbf{A}_1}) \leftarrow \mathsf{TrapGen}(n, m, q)$, and randomly choose $\mathbf{A}_{2,1}, \mathbf{A}_{2,2} \leftarrow_R \mathbb{Z}_q^{n \times m}$.
  2. Compute $(\mathbf{B}, \mathbf{T}_{\mathbf{B}}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{A}_1, \mathbf{0})$.
  3. For $j = 1, \ldots, N$, define $\bar{\mathbf{A}}_j = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2})$, extract a basis $\mathbf{T}_{\bar{\mathbf{A}}_j} \leftarrow \mathsf{ExtRndBasis}(\bar{\mathbf{A}}_j, \mathbf{T}_{\mathbf{A}_1}, s)$ such that $\|\widetilde{\mathbf{T}}_{\bar{\mathbf{A}}_j}\| \leq s\sqrt{2m}$.

4. Define the group public key $gpk = \{\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}\}$, the group manager secret key $gmsk = \mathbf{T}_\mathbf{B}$, and the group member's secret keys $\mathbf{gsk} = \{gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}\}_{j \in \{1,\ldots,N\}}$

– $\mathsf{Sign}(gpk, gsk_j, M)$: Take the group public key $gpk = \{\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}\}$, the j-th user's secret key $gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}$, and a message $M \in \{0,1\}^*$ as inputs, proceed as follows:
  1. Compute $(\mathbf{x}_1, \mathbf{x}_2) \leftarrow \mathsf{SamplePre}(\bar{\mathbf{A}}_j, \mathbf{T}_{\bar{\mathbf{A}}_j}, \beta, \mathbf{0})$, where $\mathbf{x}_1, \mathbf{x}_2 \in D_{\mathbb{Z}^m,\beta}$.
  2. Choose $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow_R \chi_\alpha$, and compute
  $$\mathbf{c} = \mathbf{B}^T\mathbf{s} + p\mathbf{e} + \mathbf{x}_1$$
  3. Generate a NIZKPoK proof π_1 of $(\mathbf{s}, \mathbf{e}, \mathbf{x}_1)$ such that $(\mathbf{B}, \mathbf{c}, \eta; \mathbf{s}, \mathbf{e}, \mathbf{x}_1) \in R_{\text{eLWE}}$.
  4. Let $\bar{\beta} := \lfloor \beta \rfloor$ and $\ell = \lceil \log_{\bar{\beta}} N \rceil$, define $\mathbf{b} = \mathbf{A}_{2,2}\mathbf{x}_2$, and $\mathbf{D} = (\mathbf{b}, \bar{\beta}\mathbf{b}, \ldots, \bar{\beta}^{\ell-1}\mathbf{b}) \in \mathbb{Z}_q^{n\times\ell}$. Generate a NIZKPoK π_2 of $\mathbf{x}_1, \mathbf{e}$, and $\mathbf{v}_j = (v_0, \ldots, v_{\ell-1}) \in \mathbb{Z}_{\bar{\beta}}^{\ell}$ of $j \in [N]$ such that,
  $$\begin{array}{c} \mathbf{A}_1\mathbf{c} + \mathbf{A}_{2,1}\mathbf{x}_2 = (p\mathbf{A}_1)\mathbf{e} - \mathbf{D}\mathbf{v}_j, \text{ and} \\ \mathbf{A}_1\mathbf{c} = (p\mathbf{A}_1)\mathbf{e} + \mathbf{A}_1\mathbf{x}_1 \end{array} \tag{2}$$
  where the challenge is computed by $H(\mathbf{c}, \mathbf{x}_2, \pi_1, M, \mathbf{Com})$, and $\mathbf{Com}$ is the commitment message for the NIZKPoK proof of π_2. (Note that the proof, *i.e.*, π_2, is actually a standard composition of our protocol in Section 4.2 and the protocol for R_{ISIS} [44,51] according to the nice property of the underlying Σ-protocol. More discussions are given in the the appendix A.)
  5. Output the signature $\sigma = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2)$.

– $\mathsf{Verify}(gpk, M, \sigma)$: Parse $\sigma = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2)$, return 1 if $\|\mathbf{x}_2\| \le \beta\sqrt{m}$, $\mathbf{A}_{2,2}\mathbf{x}_2 \ne \mathbf{0}$, and the proofs π_1, π_2 are valid, else return 0.

– $\mathsf{Open}(gpk, gmsk, M, \sigma)$: Parse $gpk = \{\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}\}$ and $gmsk = \mathbf{T}_\mathbf{B}$, compute $\mathbf{x}_1$ by decrypting $\mathbf{c}$ using $\mathbf{T}_\mathbf{B}$. Then, compute $\mathbf{y}_0 = \mathbf{A}_{2,2}\mathbf{x}_2$ and $\mathbf{y}_1 = -\mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_{2,1}\mathbf{x}_2$. If $\mathbf{y}_0 \ne \mathbf{0}$ and there is a $j \in \mathbb{Z}_q^*$ such that $\mathbf{y}_1 = j \cdot \mathbf{y}_0 \mod q$, output j, else output $\perp$.

Remark 1. One can decrypt $\mathbf{c}$ by first computing $\mathbf{T}_\mathbf{B}^T \cdot \mathbf{c} = \mathbf{T}_\mathbf{B}^T(p\mathbf{e} + \mathbf{x}_1) \mod q$. If $\|\mathbf{T}_\mathbf{B}^T(p\mathbf{e} + \mathbf{x}_1)\|_\infty < q/2$, one can expect that $\mathbf{T}_\mathbf{B}^T(p\mathbf{e} + \mathbf{x}_1) = (\mathbf{T}_\mathbf{B}^T\mathbf{c} \mod q)$ holds over $\mathbb{Z}$. Thus, $\hat{\mathbf{x}} = (p\mathbf{e} + \mathbf{x}_1)$ can be solved by using Gaussian elimination over $\mathbb{Z}$ since $\mathbf{T}_\mathbf{B} \in \mathbb{Z}^{m\times m}$ is full-rank. Finally, $\mathbf{x}_1 = \hat{\mathbf{x}} \mod p$ can be successfully recovered if $\|\mathbf{x}_1\|_\infty < p/2$.

For the correctness of our group signature scheme, we have the following theorem.

Theorem 5. *Assume n is the security parameter, and all other parameters $m, s, \alpha, \beta, \eta, p, q$ are functions of n defined as in (1), where p, q are primes. Then, the group signature $\mathcal{GS}$ is correct, and the group public key and the signature have bit-length $4nm\log q$ and $O(tm\log q)$, respectively, where $t = \omega(\log n)$.*

Proof. Since we set $m = 6n^{1+\delta} > \lceil 6n \log q + n \rceil$, the two algorithms $\mathsf{TrapGen}$ and $\mathsf{SuperSamp}$ can work correctly with overwhelming probability. In particular, we have $\|\widetilde{\mathbf{T}}_{\mathbf{A}_1}\| \leq O(\sqrt{m})$, and $\|\mathbf{T}_{\mathbf{B}}\| \leq m^{1.5} \cdot O(\sqrt{\log m})$ by Proposition 3 and Proposition 6. By Proposition 4, we have $\|\widetilde{\mathbf{T}}_{\bar{\mathbf{A}}_j}\| \leq s\sqrt{2m}$ for all $i \in \{1, \ldots, N\}$ with overwhelming probability. Since the group public key only contains four matrices over $\mathbb{Z}_q^{n \times m}$, it has bit-size $4nm \log q = O(nm \log q)$.

For the Sign algorithm, since $\beta = s\sqrt{2m} \cdot \omega(\sqrt{\log 2m}) \geq \|\widetilde{\mathbf{T}}_{\bar{\mathbf{A}}_j}\| \cdot \omega(\sqrt{\log 2m})$, we have $\mathbf{x}_1, \mathbf{x}_2 \backsim D_{\mathbb{Z}^m, \beta}$ by the correctness of the $\mathsf{SamplePre}$ algorithm in [37], and $\|\mathbf{x}_i\| \leq \beta\sqrt{m}$ with overwhelming probability. In addition, since $\mathbf{e}$ is chosen from χ_α, we have $\|\mathbf{e}\| \leq \alpha q \sqrt{m}$ with overwhelming probability. By the choices of $\eta = \max(\beta, \alpha q)\sqrt{m}$, the algorithm can successfully generate the proofs π_1 and π_2. For the bit-length of the signature $\sigma = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2)$, we know that both the bit-length of $\mathbf{c}$ and $\mathbf{x}_2$ are at most $m \log q$. In addition, if we set the repetition parameter $t = \omega(\log n)$ for the proof π_1 and π_2, the bit-length of π_1 and π_2 are at most $(3m - n)t \log q$ and $(2m + 2n + \ell)t \log q$, respectively. Thus, the total bit-length of the signature σ is less than $2m \log q + (5m + n + \ell)t \log q = O(t(m + \log N) \log q) = O(tm \log q)$ since $\ell = \lceil \log_{\bar{\beta}} N \rceil \ll n$ and $m = O(n \log q)$.

Note that $\mathbf{x}_2 \backsim D_{\mathbb{Z}^m, \beta}$,[1] therefore $\Pr[\mathbf{A}_{2,2}\mathbf{x}_2 = \mathbf{0}] \leq O(q^{-n})$ by Lemma 3. Moreover, by the completeness of π_1 and π_2, the algorithm Verify will work correctly with overwhelming probability. As for the Open algorithm, we only have to show that we can correctly decrypt $\mathbf{x}_1$ from $\mathbf{c}$ by using $\mathbf{T}_{\mathbf{B}}$. Since $\mathbf{T}_{\mathbf{B}}^T \cdot \mathbf{c} = \mathbf{T}_{\mathbf{B}}^T(p\mathbf{e} + \mathbf{x}_1) \mod q$ holds, one can expect that $\mathbf{T}_{\mathbf{B}}^T(p\mathbf{e} + \mathbf{x}_1) = (\mathbf{T}_{\mathbf{B}}^T\mathbf{c} \mod q)$ holds over $\mathbb{Z}$ if $\|\mathbf{T}_{\mathbf{B}}^T(p\mathbf{e} + \mathbf{x}_1)\|_\infty < q/2$. Thus, one can solve $\hat{\mathbf{x}} = (p\mathbf{e} + \mathbf{x}_1)$ by Gaussian elimination over $\mathbb{Z}$ since $\mathbf{T}_{\mathbf{B}} \in \mathbb{Z}^{m \times m}$ is full-rank. Moreover, by the choices of p and β, we have $\|\mathbf{x}_1\|_\infty \leq \|\mathbf{x}_1\| < p$, therefore $\mathbf{x}_1$ can be recovered by computing $\hat{\mathbf{x}} \mod p$. We finish this proof by showing that $\|\mathbf{T}_{\mathbf{B}}^T(p\mathbf{e} + \mathbf{x}_1)\|_\infty < q/2$. Actually, by Lemma 2 and Lemma 1, we have $\|p\mathbf{e} + \mathbf{x}_1\| \leq 3m^6 \cdot \omega(\log^3 m)$. By Proposition 6, we have $\|\mathbf{T}_{\mathbf{B}}\| \leq m^{1.5} \cdot \omega(\sqrt{\log m})$. It is easy to check that $\|\mathbf{T}_{\mathbf{B}}^T(p\mathbf{e} + \mathbf{x}_1)\|_\infty \leq 3m^8 \cdot \omega(\log^{3.5} m) \ll q$, which satisfies the requirement. $\square$

5.2 The Security.

For security, namely CPA-anonymity and full traceability, we have the following two theorems.

Theorem 6 (CPA-Anonymity). *Under the LWE assumption, our group signature $\mathcal{GS}$ is CPA-anonymous in the random oracle model.*

Proof. We prove Theorem 6 via a sequence of games.

In game G_0, the challenger honestly generates the group public key $gpk = \{\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}\}$, the group manager secret key $gmsk = \mathbf{T}_{\mathbf{B}}$, and the group member's

[1] As noted by Cash *et al.* [31], the output distribution of the $\mathsf{SamplePre}$ algorithm in [37] is statistically close to the distribution $(\mathbf{x}_1, \mathbf{x}_2)$ that samples as follows: randomly choose $\mathbf{x}_2 \leftarrow_R D_{\mathbb{Z}^m, s}$, and then compute $\mathbf{x}_1$ using $\mathbf{T}_{\mathbf{A}_1}$ to satisfy the condition $\mathbf{A}_1\mathbf{x}_1 + (\mathbf{A}_{2,1} + j\mathbf{A}_{2,2})\mathbf{x}_2 = \mathbf{0}$. We note that this is also the reason why we do not encrypt $\mathbf{x}_2$ as in [44], since it leaks little information about j without $\mathbf{x}_1$.

secret key $\mathbf{gsk} = \{gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}\}_{j\in\{1,\dots,N\}}$ by running the KeyGen algorithm. Then, it gives $(gpk, \mathbf{gsk})$ to the adversary $\mathcal{A}$, and obtains a message M, and two user indexes $i_0, i_1 \in \{1,\dots,N\}$. Finally, the challenger randomly chooses a bit $b \leftarrow_R \{0,1\}$, computes $\sigma^* = (\mathbf{c}^*, \mathbf{x}_2^*, \pi_1^*, \pi_2^*) \leftarrow \mathsf{Sign}(gpk, gsk_{i_b}, M)$, and returns σ^* to $\mathcal{A}$.

In Game G_1, the challenger behaves almost the same as in G_0, except that it uses the NIZKPoK simulators (by appropriately programming the random oracle) to generate π_1^*, π_2^*. By the property of the NIZKPoKs, G_1 is computationally indistinguishable from G_0.

In Game G_2, the challenger behaves almost the same as in G_1, except that it first chooses $\mathbf{x}_2^*$ from $D_{\mathbb{Z}^m,\beta}$, and then uses $\mathbf{T}_{\mathbf{A}_1}$ to extract $\mathbf{x}_1^*$ such that $\bar{\mathbf{A}}_{i_b}(\mathbf{x}_1^*;\mathbf{x}_2^*) = \mathbf{0}$. By the property of the $\mathsf{SamplePre}$ algorithm from [37,31], Game G_2 is statistically close to Game G_1

In Game G_3, the challenger behaves almost the same as in G_2, except that it computes $\mathbf{c}^* = \mathbf{u} + \mathbf{x}_1^*$ with a randomly chosen $\mathbf{u} \leftarrow_R \mathbb{Z}_q^m$.

Lemma 6. *Under the LWE assumption, Game G_3 is computationally indistinguishable from Game G_2.*

Proof. Assume there is an algorithm $\mathcal{A}$ which distinguishes G_2 from G_3 with non-negligible probability. Then, there is an algorithm $\mathcal{B}$ that breaks the LWE assumption. Formally, given a LWE tuple $(\hat{\mathbf{B}}, \hat{\mathbf{u}}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, $\mathcal{B}$ sets $\mathbf{B} = p\hat{\mathbf{B}}$, and computes $(\mathbf{A}_1, \mathbf{T}_{\mathbf{A}_1}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{B}, \mathbf{0})$. Then, it chooses $\mathbf{A}_{2,1}, \mathbf{A}_{2,2} \leftarrow_R \mathbb{Z}_q^{n\times m}$. For $j = 1,\dots,N$, define $\bar{\mathbf{A}}_j = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2})$, extract a random basis $\mathbf{T}_{\bar{\mathbf{A}}_j} \leftarrow \mathsf{ExtRndBasis}(\bar{\mathbf{A}}_j, \mathbf{T}_{\mathbf{A}_1}, s)$. Finally, $\mathcal{B}$ gives the group public key $gpk = \{\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}\}$, and the group members' secret keys $\mathbf{gsk} = \{gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}\}_{j\in\{1,\dots,N\}}$ to $\mathcal{A}$. Note that the distributions of $gpk, \mathbf{gsk}$ are statistically close to that in Game G_2 and G_3 by Proposition 4.

When generating the challenge signature, $\mathcal{B}$ behaves the same as the challenger in G_2, except that it computes $\mathbf{c} = p\hat{\mathbf{u}} + \mathbf{x}_1^*$. We note that if $(\hat{\mathbf{B}}, \hat{\mathbf{u}})$ is a LWE tuple with respect to the error distribution χ_α, $\mathbf{c}$ is the same as in G_2. Otherwise, we have that $p\hat{\mathbf{u}}$ is uniformly distributed over $\mathbb{Z}_q^m$ (since p, q are primes, and $p < q$), which shows that $\mathbf{c}$ has the same distribution as in G_3. If $\mathcal{A}$ can distinguish G_2 from G_3 with advantage ϵ, then $\mathcal{B}$ can break the LWE assumption with advantage $\epsilon - negl(k)$. □

In Game G_4, the challenger behaves almost the same as in G_3, except that it randomly chooses $\mathbf{c}^* \leftarrow_R \mathbb{Z}_q^m$.

Lemma 7. *In Game G_4, the probability that $b' = b$ is exactly $1/2$.*

Proof. The claim follows from the fact that the signature σ^* in G_4 is independent from the choice of i_b. □

Theorem 7 (Traceability). *Under the SIS assumption, our group signature $\mathcal{GS}$ is fully traceable in the random oracle model.*

Proof. Assume that there is an adversary $\mathcal{A}$ that breaks the full traceability of $\mathcal{GS}$, we construct an algorithm $\mathcal{B}$ breaking the SIS assumption. Formally, $\mathcal{B}$ is given a matrix $\hat{\mathbf{A}} \in \mathbb{Z}_q^m$ and tries to find a solution $\hat{\mathbf{x}} \in \mathbb{Z}_q^m$ such that $\|\hat{\mathbf{x}}\| \le poly(m)$ and $\hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{0}$.

Setup. $\mathcal{B}$ randomly chooses $\mathbf{R} \leftarrow_R \{-1,1\}^{m\times m}$ and $j^* \leftarrow_R \{-4m^{2.5}N+1,\ldots,4m^{2.5}N-1\}$, and computes $(\mathbf{A}_{2,2},\mathbf{T}_{\mathbf{A}_{2,2}}) \leftarrow \mathsf{TrapGen}(n,m,q)$. Then, it sets $\mathbf{A}_1 = \hat{\mathbf{A}}$ and $\mathbf{A}_{2,1} = \mathbf{A}_1\mathbf{R} - j^*\mathbf{A}_{2,2}$. Finally, compute $(\mathbf{B},\mathbf{T}_\mathbf{B}) \leftarrow \mathsf{SuperSamp}(n,m,q,\mathbf{A}_1,\mathbf{0})$, and give the group public key $gpk = \{\mathbf{A}_1,\mathbf{A}_{2,1},\mathbf{A}_{2,2},\mathbf{B}\}$, and the group manager secret key $gmsk = \mathbf{T}_\mathbf{B}$ to the adversary $\mathcal{A}$.

Secret Key Queries. Upon receiving the secret key query for user j from $\mathcal{A}$, $\mathcal{B}$ aborts if $j = j^*$ or $j \notin \{1,\ldots,N\}$. Otherwise, it defines $\bar{\mathbf{A}}_j = (\mathbf{A}_1\|\mathbf{A}_{2,1}+j^*\mathbf{A}_{2,2}) = (\mathbf{A}_1\|\mathbf{A}_1\mathbf{R}+(j-j^*)\mathbf{A}_{2,2})$, extracts a random basis $\mathbf{T}_{\bar{\mathbf{A}}_j} \leftarrow \mathsf{ExtBasisRight}(\bar{\mathbf{A}}_j,\mathbf{R},\mathbf{T}_{\mathbf{A}_{2,2}},s)$,[2] and returns it to $\mathcal{A}$.

Sign Queries. Upon receiving a signing query for message M under user j from $\mathcal{A}$, $\mathcal{B}$ returns $\perp$ if $j \notin \{1,\ldots,N\}$. Else if $j = j^*$, $\mathcal{B}$ generates a signature on M using the NIZKPoK simulators for π_1 and π_2 (*i.e.*, by simply choosing $\mathbf{c} \leftarrow_R \mathbb{Z}_q^m$ and $\mathbf{x}_2 \leftarrow_R D_{\mathbb{Z}^m,s}$). Otherwise, it generates the signature by first extracting the j-th user's secret key as in answering the secret key queries.

Forge. Upon receiving a forged valid signature $\sigma = (\mathbf{c},\mathbf{x}_2,\pi_1,\pi_2)$ with probability ϵ, $\mathcal{B}$ extracts the knowledge $\mathbf{e},\mathbf{x}_1$ and $\mathbf{v}_j$ with norm at most $4\eta m^2$ by programming the random oracle twice to generate two different "challenges". By the forking lemma of [12], $\mathcal{B}$ can succeed with probability at least $\epsilon(\epsilon/q_h - 2^{-t})$, where q_h is the maximum number of hash queries of $\mathcal{A}$. Then, $\mathcal{B}$ decrypts $\mathbf{c}$ using $\mathbf{T}_\mathbf{B}$, and obtains $(\mathbf{e}',\mathbf{x}_1')$, and distinguishes the following cases:

- If $(\mathbf{x}_1',\mathbf{e}') \neq (\mathbf{x}_1,\mathbf{e})$, we have $\mathbf{A}_1\mathbf{c} = p\mathbf{A}_1\mathbf{e} + \mathbf{A}_1\mathbf{x}_1 = p\mathbf{A}_1\mathbf{e}' + \mathbf{A}_1\mathbf{x}_1'$. Thus, $\hat{\mathbf{x}} = p(\mathbf{e}-\mathbf{e}')+(\mathbf{x}_1-\mathbf{x}_1')$ is a solution of the SIS problem, $\mathcal{B}$ returns $\hat{\mathbf{x}}$ as its own solution. Note that in this case, we have $\|\hat{\mathbf{x}}\| \leq 8(p+1)\eta m^2 = m^8\cdot\omega(\log^3 m)$.
- Otherwise, if $(\mathbf{x}_1',\mathbf{e}') = (\mathbf{x}_1,\mathbf{e})$, we have $\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_{2,1}\mathbf{x}_2 + j\mathbf{A}_{2,2}\mathbf{x}_2 = \mathbf{0}$ according to equation (2) of π_2, where $j = \sum_{i=0}^{\ell-1} v_i\bar{\beta}^i$ and $\mathbf{v}_h = (v_0,\ldots,v_{\ell-1})$. A simple calculation indicates that $|j| < 4m^{2.5}N < q$ (we note that $\|\mathbf{v}_h\| \leq 4\eta m^2 = 4\beta m^{2.5}$). Since $\mathbf{A}_{2,2}\mathbf{x}_2 \neq \mathbf{0}$ and q is a prime, the open algorithm will always output j, namely, it will never output $\perp$. In addition, if $j \neq j^*$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ returns $\hat{\mathbf{x}} = \mathbf{x}_1 + \mathbf{R}\mathbf{x}_2$ as its own solution.
  Since j^* is randomly chosen from $\{-4m^{2.5}N+1,\ldots,4m^{2.5}N-1\}$, the probability that $j^* = j$ is at least $\frac{1}{8m^{2.5}N}$. Conditioned on $j^* = j$, we have $\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_{2,1}\mathbf{x}_2 + j\mathbf{A}_{2,2}\mathbf{x}_2 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_1\mathbf{R}\mathbf{x}_2 = \mathbf{0}$, which shows that $\hat{\mathbf{x}} = \mathbf{x}_1 + \mathbf{R}\mathbf{x}_2$ is a solution of the SIS problem, in particular, we have $\|\hat{\mathbf{x}}\| \leq \eta m^{2.5}\cdot\omega(\sqrt{\log m}) = m^{4.5}\omega(\log^2 m)$ by [2, Lem. 5].

In all, the probability that $\mathcal{B}$ solves the SIS problem is at least $\frac{\epsilon(\epsilon/q_h-2^{-t})}{8m^{2.5}N}$, which is non-negligible if ϵ is non-negligible. Moreover, since the norm of $\hat{\mathbf{x}}$ is at most $m^8\cdot\omega(\log^3 m)$ and $q \geq m^{8.5}\cdot\omega(\log^{2.5} m)$, we have that the security of our scheme is based on the hardness of the SIVP problem in the worst case to within a polynomial approximation factor, by Proposition 2. □

6 Achieving Full Anonymity

In this section, we present a group signature scheme with CCA-anonymity. First, we introduce a primitive called full-rank differences function.

[2] Recall that $\mathbf{T}_{\mathbf{A}_{2,2}}$ is also a short basis of $\mathbf{\Lambda}_q^\perp(j\cdot\mathbf{A}_{2,2})$ for all $j \neq 0 \mod q$.

Definition 5 ([2]). *Let q be a prime and n be a positive integer. We say that a function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is an encoding with full-rank differences (FRD) if:*

- *for any $x \neq y$, the matrix $H(x) - H(y) \in \mathbb{Z}_q^{n\times n}$ is full rank;*
- *H is computable in polynomial time in $n \log q$.*

Let $\mathcal{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$ be a one-time strongly existentially unforgeable (ot-sEUF) signature scheme (please refer to Appendix B for a formal definition), and $H : \{0,1\}^* \to \mathbb{Z}_q^{n\times n}$ be an encoding with full-rank differences. Assume that the security parameter is n, and δ is a real such that $n^{1+\delta} > \lceil (n+1)\log q + n \rceil$, all other parameters $m, s, \alpha, \beta, \eta, p, q$ are determined as follows:

$$\begin{aligned}
&m = 6n^{1+\delta}\\
&s = m \cdot \omega(\log m)\\
&\beta = s\sqrt{3m} \cdot \omega(\sqrt{\log 3m}) = m^{1.5} \cdot \omega(\log^{1.5} m)\\
&p = m^{2.5}\beta = m^4 \cdot \omega(\log^{1.5} m)\\
&q = m^{2.5} \cdot \max(pm^{2.5} \cdot \omega(\log m), 4N) = m^{2.5} \max(m^{6.5} \cdot \omega(\log^{2.5} m), 4N)\\
&\alpha = 2\sqrt{m}/q\\
&\eta = \max(\beta, \alpha q \sqrt{m})\sqrt{2m} = m^2 \cdot \omega(\log^{1.5} m)
\end{aligned} \qquad (3)$$

Now, we extend our basic group signature $\mathcal{GS}$ to $\mathcal{GS}' = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ by replacing the underlying encryption with a CCA one:

- $\mathsf{KeyGen}(1^n, 1^N)$: Take the security parameter n and the maximum number of group members N as inputs, set an integer $m \in \mathbb{Z}$, primes $p, q \in \mathbb{Z}$, and $s, \alpha, \beta, \eta \in \mathbb{R}$ as in (3), and choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$ (modeled as random oracle) for the NIZKPoK proof, where $t = \omega(\log n)$. Then, the algorithm proceeds as follows:
  1. Compute $(\mathbf{A}_{1,1}, \mathbf{T}_{\mathbf{A}_{1,1}}) \leftarrow \mathsf{TrapGen}(n, m, q)$, and randomly choose $\mathbf{A}_{1,2}$, $\mathbf{A}_{2,1}, \mathbf{A}_{2,2} \leftarrow_R \mathbb{Z}_q^{n\times m}$. Denote $\mathbf{A}_1 = (\mathbf{A}_{1,1} \| \mathbf{A}_{1,2}) \in \mathbb{Z}_q^{n\times 2m}$.
  2. Randomly choose $\mathbf{B}_{2,1} \leftarrow_R \mathbb{Z}_q^{n\times m}$, compute $(\mathbf{B}_1, \mathbf{T}_{\mathbf{B}_1}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{A}_{1,1}, -\mathbf{A}_{1,2}\mathbf{B}_{2,1}^T)$ and $(\mathbf{B}_{2,2}, \mathbf{T}_{\mathbf{B}_{2,2}}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{A}_{1,2}, \mathbf{0})$. It is easy to check that for any matrix $\mathbf{C}$, we have $\mathbf{A}_1(\mathbf{B}_1 \| \mathbf{B}_{2,1} + \mathbf{C}\mathbf{B}_{2,2})^T = \mathbf{0}$.
  3. For $j = 1, \ldots, N$, define $\bar{\mathbf{A}}_j = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2}) \in \mathbb{Z}_q^{n\times 3m}$, extract a random basis $\mathbf{T}_{\bar{\mathbf{A}}_j} \leftarrow \mathsf{ExtRndBasis}(\bar{\mathbf{A}}_j, \mathbf{T}_{\mathbf{A}_{1,1}}, s)$.
  4. Define the group public key $gpk = \{\mathbf{A}_{1,1}, \mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}_1, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}\}$, the group manager secret key $gmsk = \mathbf{T}_{\mathbf{B}_1}$, and the group member's secret key $\mathbf{gsk} = \{gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}\}_{j\in\{1,\ldots,N\}}$.
- $\mathsf{Sign}(gpk, gsk_j, M)$: Take the group public key $gpk = \{\mathbf{A}_{1,1}, \mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}_1, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}\}$, the j-th user's secret key $gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}$, and a message $M \in \{0,1\}^*$, proceed as follows:
  1. Compute $(\mathbf{x}_1, \mathbf{x}_2) \leftarrow \mathsf{SamplePre}(\bar{\mathbf{A}}_j, \mathbf{T}_{\bar{\mathbf{A}}_j}, \mathbf{0}, \beta)$, where $\mathbf{x}_1 \in D_{\mathbb{Z}^{2m},\beta}, \mathbf{x}_2 \in D_{\mathbb{Z}^m,\beta}$.
  2. Generate a one-time signature verification and signing keys $(svk, ssk) \leftarrow \mathcal{SIG}.\mathsf{KeyGen}(1^k)$, denote $\hat{\mathbf{B}}_{svk} = (\mathbf{B}_1 \| \mathbf{B}_{2,1} + H(svk)\mathbf{B}_{2,2}) \in \mathbb{Z}_q^{n\times 2m}$. Randomly choose $\mathbf{s} \leftarrow_R \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow_R \chi_\alpha^m$, and $\mathbf{R} \leftarrow_R \{0,1\}^{m\times m}$, compute

$$\mathbf{c} = \hat{\mathbf{B}}_{svk}^T \mathbf{s} + p\hat{\mathbf{e}} + \mathbf{x}_1, \text{ where } \hat{\mathbf{e}} = \begin{pmatrix} \mathbf{e} \\ \mathbf{R}^T\mathbf{e} \end{pmatrix}$$

3. Generate a NIZKPoK proof π_1 of $(\hat{\mathbf{e}}, \mathbf{x}_1)$ so that $(\hat{\mathbf{B}}_{svk}, \mathbf{c}, \eta; \mathbf{s}, \hat{\mathbf{e}}, \mathbf{x}_1) \in R_{\text{eLWE}}$.
4. Let $\bar{\beta} := \lfloor \beta \rfloor$ and $\ell = \lceil \log_{\bar{\beta}} N \rceil$, define $\mathbf{b} = \mathbf{A}_{2,2}\mathbf{x}_2$, and $\mathbf{D} = (\mathbf{b}, \bar{\beta}\mathbf{b}, \ldots, \bar{\beta}^{\ell-1}\mathbf{b}) \in \mathbb{Z}_q^{n \times \ell}$. Generate a NIZKPoK π_2 of $\mathbf{x}_1, \hat{\mathbf{e}} = (\mathbf{e}; \mathbf{R}^T\mathbf{e})$, and $\mathbf{v}_j = (v_0, \ldots, v_{\ell-1}) \in \mathbb{Z}_{\bar{\beta}}^{\ell}$ of $j \in [N]$ such that,
$$\begin{gathered} \mathbf{A}_1\mathbf{c} + \mathbf{A}_{2,1}\mathbf{x}_2 = (p\mathbf{A}_1)\hat{\mathbf{e}} - \mathbf{D}\mathbf{v}_j, \text{ and} \\ \mathbf{A}_1\mathbf{c} = (p\mathbf{A}_1)\hat{\mathbf{e}} + \mathbf{A}_1\mathbf{x}_1 \end{gathered} \tag{4}$$
where the challenge is computed by $H(svk, \mathbf{c}, \mathbf{x}_2, \pi_1, M, \mathbf{Com})$, and $\mathbf{Com}$ is the commitment message for the NIZKPoK proof of π_2.
5. Denote $\sigma_1 = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2)$, compute $\sigma_2 = \mathcal{SIG}.\mathsf{Sign}(ssk, \sigma_1)$, return the group signature $\sigma = (svk, \sigma_1, \sigma_2)$.

– $\mathsf{Verify}(gpk, M, \sigma)$: Parse $\sigma = (svk, \sigma_1 = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2), \sigma_2)$, return 1 if $\mathcal{SIG}.\mathsf{Ver}(svk, \sigma_1, \sigma_2) = 1$, $\|\mathbf{x}_2\| \leq \beta\sqrt{m}$, $\mathbf{A}_{2,2}\mathbf{x}_2 \neq \mathbf{0}$, and the proofs π_1, π_2 are valid, else return 0.
– $\mathsf{Open}(gpk, gmsk, M, \sigma)$: Parse $gpk = \{\mathbf{A}_{1,1}, \mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}_1, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}\}$ and $gmsk = \mathbf{T}_{\mathbf{B}_1}$. Denote $\hat{\mathbf{B}}_{svk} = (\mathbf{B}_1 \| \mathbf{B}_{2,1} + H(svk)\mathbf{B}_{2,2}) \in \mathbb{Z}_q^{n \times 2m}$, extract a basis $\mathbf{T}_{\hat{\mathbf{B}}_{svk}} \leftarrow \mathsf{ExtRndBasis}(\hat{\mathbf{B}}_{svk}, \mathbf{T}_{\mathbf{B}_1}, s\sqrt{m} \cdot \omega(\sqrt{\log m}))$. Compute $\mathbf{x}_1$ by decrypting $\mathbf{c}$ using $\mathbf{T}_{\hat{\mathbf{B}}_{svk}}$, and let $\mathbf{y}_1 = \mathbf{A}_{2,2}\mathbf{x}_2$ and $\mathbf{y}_2 = -\mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_{2,1}\mathbf{x}_2$. If $\mathbf{y}_1 \neq \mathbf{0}$ and there is a $j \in \mathbb{Z}_q^*$ such that $\mathbf{y}_2 = j \cdot \mathbf{y}_1$, return j, else $\perp$.

We summarize the correctness of $\mathcal{GS}'$ in the following theorem. The proof is very similar to Theorem 5, we omit the details.

Theorem 8. *Assume that n is the security parameter, and all other parameters $m, s, \alpha, \beta, \eta, p, q$ are functions of n defined as in (3), where p, q are primes. Then, the group signature $\mathcal{GS}'$ is correct, and the group public key and the signature have bit-length $O(nm \log q)$ and $O(tm \log q) + l_{\mathcal{SIG}}$, respectively, where $l_{\mathcal{SIG}}$ is the total bit-length of the verification key and signature (i.e., svk and σ_2) of the underlying one-time signature.*

6.1 Full Anonymity

Theorem 9 (Full Anonymity). *Under the LWE assumption, our group signature $\mathcal{GS}'$ is fully anonymous in the random oracle model.*

Proof. We prove the theorem via a sequence of games.

In game G_0, the challenger honestly creates the group public key $gpk = \{\mathbf{A}_{1,1}, \mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}_1, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}\}$, the group manager's secret key $gmsk = \mathbf{T}_{\mathbf{B}_1}$, and the group member's secret key $\mathbf{gsk} = \{gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}\}_{j \in \{1,\ldots,N\}}$ by running the KeyGen algorithm. Then, it gives $(gpk, \mathbf{gsk})$ to the adversary, and obtains a message M, and two user indexes i_0, i_1. Finally, the challenger chooses a bit $b \leftarrow_R \{0,1\}$, computes $\sigma^* = (svk^*, \sigma_1^* = (\mathbf{c}^*, \mathbf{x}_2^*, \pi_1^*, \pi_2^*), \sigma_2^*) \leftarrow \mathsf{Sign}(gpk, gsk_{i_b}, M)$, and returns σ^* to the adversary $\mathcal{A}$.

In Game G_1, the challenger behaves almost the same as in G_0, except that it uses the NIZKPoK simulators (by appropriately programming the random oracle) to generate π_1^*, π_2^*. By the property of the NIZKPoKs, G_1 is indistinguishable from G_0.

In Game G_2, the challenger behaves almost the same as in G_1, except that it generates a one-time signature key pair $(svk^*, ssk^*) \leftarrow \mathcal{SIG}.\mathsf{KeyGen}(1^k)$ at the beginning of the experiment, and aborts when $\mathcal{A}$ outputs a valid signature $\sigma = (svk^*, \sigma_1, \sigma_2) \neq \sigma^*$, where σ^* is the challenge signature.

Lemma 8. *If the signature $\mathcal{SIG}$ is one-time strongly existentially unforgeable (ot-sEUF) for any PPT forger, then Game G_2 is computationally indistinguishable from G_1.*

Proof. If there is an adversary $\mathcal{A}$ that outputs a valid signature $\sigma = (svk^*, \sigma_1, \sigma_2) \neq \sigma^*$, we can construct a forger $\mathcal{F}$ that breaks the strong unforgeability of $\mathcal{SIG}$. Actually, given a challenge verification key svk^* from the challenger of the ot-sEUF security game (please refer to Appendix B), $\mathcal{F}$ behaves almost the same as the challenger in G_2 to simulate the attack environment for $\mathcal{A}$, except that it uses svk^* in generating the challenge signature σ^* by making a signing query to its own signing oracle. Whenever $\mathcal{A}$ outputs a valid signature $\sigma = (svk^*, \sigma_1, \sigma_2) \neq \sigma^*$, $\mathcal{F}$ returns (σ_1, σ_2) as its own forgery. Obviously, if $\mathcal{A}$ can output a valid signature with probability ϵ, $\mathcal{F}$ can successfully forge a signature for the underlying one-time signature almost with the same probability ϵ. $\square$

In Game G_3, the challenger behaves almost the same as in G_2, except that it first chooses $\mathbf{x}_2$ from $D_{\mathbb{Z}^m,\beta}$, and then uses the trapdoor $\mathbf{T}_{\mathbf{A}_{1,1}}$ to extract $\mathbf{x}_1$ such that $\bar{\mathbf{A}}_{i_b}(\mathbf{x}_1; \mathbf{x}_2) = \mathbf{0}$. By the property of the $\mathsf{SamplePre}$ algorithm in [37,31], we have that Game G_3 is statistically close to Game G_2 .

In Game G_4, the challenger behaves almost the same as in G_3, except that it computes $\mathbf{c}^* = \mathbf{u} + \mathbf{x}_1$ for a randomly chosen $\mathbf{u} \leftarrow_R \mathbb{Z}_q^{2m}$.

Lemma 9. *Under the LWE assumption, Game G_4 is computationally indistinguishable from Game G_3.*

Proof. Assume that there is an algorithm $\mathcal{A}$ which distinguishes G_3 from G_4 with non-negligible probability. We will show that there is an algorithm $\mathcal{B}$ breaking the LWE assumption. Formally, given a LWE tuple $(\hat{\mathbf{B}}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, $\mathcal{B}$ sets $\mathbf{B}_1 = p\hat{\mathbf{B}}$. Randomly choose $\mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2} \leftarrow_R \mathbb{Z}_q^{n\times m}$ and $\mathbf{R} \in \{0,1\}^{m\times m}$, generate a one-time signature pair $(svk^*, ssk^*) \leftarrow \mathcal{SIG}.\mathsf{KeyGen}(1^k)$. Then, it computes the public parameters $(\mathbf{B}_{2,2}, \mathbf{T}_{\mathbf{B}_{2,2}}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{A}_{1,2}, \mathbf{0})$, $\mathbf{B}_{2,1} = \mathbf{B}_1\mathbf{R} - H(svk^*)\mathbf{B}_{2,2}$, and $(\mathbf{A}_{1,1}, \mathbf{T}_{\mathbf{A}_{1,1}}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{B}_1, -\mathbf{B}_{2,1}\mathbf{A}_{1,2}^T)$. Denote $\mathbf{A}_1 = (\mathbf{A}_{1,1} \| \mathbf{A}_{1,2})$.

For $j = 1, \ldots, N$, define $\bar{\mathbf{A}}_j = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2})$, extract a random basis $\mathbf{T}_{\bar{\mathbf{A}}_j} \leftarrow \mathsf{ExtRndBasis}(\bar{\mathbf{A}}_j,\ \mathbf{T}_{\mathbf{A}_{1,1}}, s)$. Finally, $\mathcal{B}$ gives the group public key $gpk = \{\mathbf{A}_{1,1}, \mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2},\ \mathbf{B}_1, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}\}$, and the group member's secret key $\mathbf{gsk} = \{gsk_j = \mathbf{T}_{\bar{\mathbf{A}}_j}\}_{j\in\{1,\ldots,N\}}$ to $\mathcal{A}$, and keeps $\mathbf{T}_{\mathbf{B}_{2,2}}$ secret. Note that the distribution of gpk, $\mathbf{gsk}$ is statistically close to that in Game G_3 and G_4.

If $\mathcal{A}$ submits a valid signature $\sigma = (svk^*, \sigma_1, \sigma_2) \neq \sigma^*$ to the open oracle, $\mathcal{B}$ aborts as in Game G_3. Otherwise, $\mathcal{B}$ defines $\hat{\mathbf{B}}_{svk} = (\mathbf{B}_1 \| \mathbf{B}_{2,1} + H(svk)\mathbf{B}_{2,2}) = (\mathbf{B}_1 \| \mathbf{B}_1\mathbf{R} + (H(svk) - H(svk^*))\mathbf{B}_{2,2})$, and extracts $\mathbf{T}_{\hat{\mathbf{B}}_{svk}} = \mathsf{ExtBasisRight}(\hat{\mathbf{B}}_{svk}, \mathbf{R}, \mathbf{T}_{\mathbf{B}_{2,2}}, s\sqrt{m}\cdot\omega(\sqrt{\log m}))$ (we note that $\mathbf{T}_{\mathbf{B}_{2,2}}$ is also a basis for $\mathbf{\Lambda}_q^{\perp}(\mathbf{C}\mathbf{B}_{2,2})$ for any full-rank $\mathbf{C} \in \mathbb{Z}_q^{n\times n}$, since $\mathbf{\Lambda}_q^{\perp}(\mathbf{C}\mathbf{B}_{2,2}) = \mathbf{\Lambda}_q^{\perp}(\mathbf{B}_{2,2})$). Finally, it uses $\mathbf{T}_{\hat{\mathbf{B}}_{svk}}$ as described in the Open algorithm to answer this query.

When generating the challenge signature, $\mathcal{B}$ works the same as in G_2, except that it computes $\mathbf{c} = p\begin{pmatrix}\mathbf{u}\\ \mathbf{R}^T\mathbf{u}\end{pmatrix} + \mathbf{x}_1$. We note that if $(\hat{\mathbf{B}}, \mathbf{u})$ is a LWE tuple for distribution χ_α, $\mathbf{c}$ is the same as in G_3. Otherwise, we show that $\begin{pmatrix}\mathbf{u}\\ \mathbf{R}^T\mathbf{u}\end{pmatrix}$ is statistically close to uniform over $\mathbb{Z}_q^{2m}$ in Lemma 10, which shows that the distribution of $\mathbf{c}$ is statistically close to that in G_4. Thus, if $\mathcal{A}$ can distinguish G_3 from G_4 with advantage ϵ, then $\mathcal{B}$ can break the LWE assumption with advantage $\epsilon - negl(k)$. This completes the proof. $\square$

Lemma 10. *Given randomly chosen* $\mathbf{B}_1 \leftarrow_R \mathbb{Z}_q^{n\times m}$ *and* $\mathbf{u} \leftarrow_R \mathbb{Z}_q^m$, *the distribution of* $(\mathbf{B}_1\mathbf{R}, \mathbf{u}, \mathbf{R}^T\mathbf{u})$ *is statistically close to uniform over* $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m \times \mathbb{Z}_q^m$, *where* $\mathbf{R}$ *is randomly chosen from* $\{-1, 1\}^{m\times m}$.

Proof. By Lemma 5, the distribution $(\mathbf{B}_1\mathbf{R}, \mathbf{u}, \mathbf{R}^T\mathbf{u})$ is statistically close to $(\mathbf{C}, \mathbf{u}, \mathbf{R}^T\mathbf{u})$, where $\mathbf{C}$ is randomly chosen from $\mathbb{Z}_q^{n\times m}$. Since the hash function defined by inner-product is a family of universal hash functions we have that the statistical distance between $(\mathbf{u}, \mathbf{R}^T\mathbf{u})$ and $(\mathbf{u}, \mathbf{v})$ for uniformly chosen $\mathbf{v} \in \mathbb{Z}_q^m$ is at most $\frac{1}{2}\sqrt{2^{-m^2+m\log q}}$ by Lemma 4, which is negligible in n if $m \geq 2n + \log q$ (thus, the choice of $m = \Omega(n \log q)$ suffices). This shows that $(\mathbf{B}_1\mathbf{R}, \mathbf{u}, \mathbf{R}^T\mathbf{u})$ is statistically close to $(\mathbf{C}, \mathbf{u}, \mathbf{v})$. $\square$

In Game G_5, the challenger behaves almost the same as in G_1, except that it randomly chooses $\mathbf{c}^* \leftarrow_R \mathbb{Z}_q^{2m}$.

Lemma 11. *In Game* G_5, *the probability that* $b' = b$ *is exactly* $1/2$.

Proof. This lemma follows from the fact that the signature σ^* in G_5 is independent from the choice of i_b. $\square$

6.2 Traceability

Theorem 10 (Traceability). *Under the SIS assumption, our group signature* $\mathcal{GS}'$ *is fully traceable in the random oracle model.*

Proof. Assume that there is an adversary $\mathcal{A}$ that breaks the full traceability of $\mathcal{GS}'$, we construct an algorithm $\mathcal{B}$ that breaks the SIS assumption. Formally, $\mathcal{B}$ is given a SIS matrix $\hat{\mathbf{A}} \in \mathbb{Z}_q^m$ and tries to find a solution $\hat{\mathbf{x}} \in \mathbb{Z}_q^m$ such that $\|\hat{\mathbf{x}}\| \leq poly(m)$ and $\hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{0}$.

Setup. The algorithm $\mathcal{B}$ sets $\mathbf{A}_{1,1} = \hat{\mathbf{A}}$. Generate $(\mathbf{A}_{2,2}, \mathbf{T}_{\mathbf{A}_{2,2}}) \leftarrow \mathsf{TrapGen}(n, m, q)$. Randomly choose $\mathbf{R}_1, \mathbf{R}_2 \leftarrow_R \{-1, 1\}^{m\times m}$ and $j^* \leftarrow_R \{-4m^{2.5}N + 1, \ldots, 4m^{2.5}N - 1\}$, and compute $\mathbf{A}_{1,2} = \mathbf{A}_{1,1}\mathbf{R}_1$, $\mathbf{A}_{2,1} = \mathbf{A}_{1,1}\mathbf{R}_2 - j^*\mathbf{A}_{2,2}$. Let $\mathbf{A}_1 = (\mathbf{A}_{1,1}\|\mathbf{A}_{1,2}) \in \mathbb{Z}_q^{n\times 2m}$. Randomly choose $\mathbf{B}_{2,1} \leftarrow_R \mathbb{Z}_q^{n\times m}$, and compute $(\mathbf{B}_1, \mathbf{T}_{\mathbf{B}_1}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{A}_{1,1}, -\mathbf{A}_{1,2}\mathbf{B}_{2,1}^T)$. Then, compute $(\mathbf{B}_{2,2}, \mathbf{T}_{\mathbf{B}_{2,2}}) \leftarrow \mathsf{SuperSamp}(n, m, q, \mathbf{A}_{1,2}, \mathbf{0})$. Give the group public key $gpk = \{\mathbf{A}_{1,1}, \mathbf{A}_{1,2}, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}, \mathbf{B}_1, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}\}$, the group manager's secret key $gmsk = \mathbf{T}_{\mathbf{B}_1}$ to the adversary $\mathcal{A}$.

Secret Key Queries. Upon receiving the secret key query for user j from $\mathcal{A}$, $\mathcal{B}$ aborts if $j = j^*$ or $j \notin \{1, \ldots, N\}$. Otherwise, define $\bar{\mathbf{A}}_j = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2}) = (\mathbf{A}_1 \| \mathbf{A}_{2,1}\mathbf{R}_2 + (j - j^*)\mathbf{A}_{2,2})$, return the secret key $\mathbf{T}_{\bar{\mathbf{A}}_j} \leftarrow \mathsf{ExtBasisRight}(\bar{\mathbf{A}}_j, \mathbf{R}_2, \mathbf{T}_{\mathbf{A}_{2,2}}, s)$ to $\mathcal{A}$.

Sign Queries. Upon receiving a signing query for message M under user j from $\mathcal{A}$, $\mathcal{B}$ returns $\perp$ if $j \notin \{1, \ldots, N\}$. Else if $j = j^*$, $\mathcal{B}$ generates a signature on M using the NIZKPoK simulators for π_1 and π_2. Otherwise, it generates the signature by first extracting the corresponding key as in answering the secret key queries.

Forge. Upon receiving a forged valid signature $\sigma = (svk, \sigma_1 = (\mathbf{c}, \mathbf{x}_2, \pi_1, \pi_2), \sigma_2)$ with probability ϵ, $\mathcal{B}$ extracts the knowledge $\hat{\mathbf{e}}, \mathbf{x}_1$ and $\mathbf{v}_j$ with normal at most $4\eta m^2$ by programming the random oracle twice to generate two different "challenges". By the forking lemma of [12], $\mathcal{B}$ can succeed with probability at least $\epsilon(\epsilon/q_h - 2^{-t})$, where h is the maximum number of hash queries of $\mathcal{A}$. Then, $\mathcal{B}$ decrypts $\mathbf{c}$ to obtain $\hat{\mathbf{e}}', \mathbf{x}_1'$, and distinguishes the following two cases:

- If $(\mathbf{x}_1', \hat{\mathbf{e}}') \neq (\mathbf{x}_1, \hat{\mathbf{e}})$, we have $\mathbf{A}_1\mathbf{c} = p\mathbf{A}_1\hat{\mathbf{e}}' + \mathbf{A}_1\mathbf{x}_1' = p\mathbf{A}_1\hat{\mathbf{e}} + \mathbf{A}_1\mathbf{x}_1$. Let $\mathbf{y} = (\mathbf{y}_1; \mathbf{y}_2) = p(\hat{\mathbf{e}} - \hat{\mathbf{e}}') + (\mathbf{x}_1 - \mathbf{x}_1')$, we have $\mathbf{0} = \mathbf{A}_1\mathbf{y} = \mathbf{A}_{1,1}\mathbf{y}_1 + \mathbf{A}_{1,2}\mathbf{y}_2 = \mathbf{A}_{1,1}(\mathbf{y}_1 + \mathbf{R}_1\mathbf{y}_2)$. Namely, $\hat{\mathbf{x}} = \mathbf{y}_1 + \mathbf{R}_1\mathbf{y}_2$ is a solution of the SIS problem, $\mathcal{B}$ returns $\hat{\mathbf{x}}$ as its own solution. Note that in this case, we have $\|\hat{\mathbf{x}}\| \leq (p+1)\eta m^{2.5} \cdot \omega(\sqrt{\log m}) = m^{8.5}\omega(\log^{3.5} m)$.
- Else if $(\mathbf{x}_1', \hat{\mathbf{e}}') = (\mathbf{x}_1, \hat{\mathbf{e}})$, we have $\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_{2,1}\mathbf{x}_2 + j\mathbf{A}_{2,2}\mathbf{x}_2 = \mathbf{0}$ according to equation (4) of π_2, where $j = \sum_{i=0}^{\ell-1} v_i\eta^i$ and $\mathbf{v}_j = (v_0, \ldots, v_{\ell-1})$. A simple calculation indicates that $|j| < 4m^{2.5}N < q$. Since $\mathbf{A}_{2,2}\mathbf{x}_2 \neq \mathbf{0}$ and q is a prime, the open algorithm will always output j, namely, it will never output $\perp$. In addition, if $j \neq j^*$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ returns $\hat{\mathbf{x}} = \mathbf{x}_{1,1} + \mathbf{R}_1\mathbf{x}_{1,2} + \mathbf{R}_2\mathbf{x}_2$ as its own solution, where $\mathbf{x}_1 = (\mathbf{x}_{1,1}; \mathbf{x}_{1,2})$.
  Since j^* is randomly chosen from $\{-4m^{2.5}N + 1, \ldots, 4m^{2.5}N - 1\}$, we have the probability that $j^* = j$ is at least $\frac{1}{8m^{2.5}N}$. Conditioned on $j^* = j$, we have $\mathbf{A}_{1,1}\mathbf{x}_{1,1} + \mathbf{A}_{1,2}\mathbf{x}_{1,2} + \mathbf{A}_{2,1}\mathbf{x}_2 + j\mathbf{A}_{2,2}\mathbf{x}_2 = \mathbf{A}_{1,1}(\mathbf{x}_{1,1} + \mathbf{R}_1\mathbf{x}_{1,2} + \mathbf{R}_2\mathbf{x}_2) = \mathbf{0}$, which shows that $\hat{\mathbf{x}} = \mathbf{x}_{1,1} + \mathbf{R}_1\mathbf{x}_{1,2} + \mathbf{R}_2\mathbf{x}_2$ is a solution of the SIS problem. In particular, we have $\|\hat{\mathbf{x}}\| \leq \eta m^{2.5} \cdot \omega(\sqrt{\log m}) = m^{4.5} \cdot \omega(\log^2 m)$ by [2, Lem. 5].

In all, the probability that $\mathcal{B}$ solves the SIS problem is at least $\frac{\epsilon(\epsilon/q_h - 2^{-t})}{8m^{2.5}N}$, which is non-negligible if ϵ is non-negligible. Moreover, the norm of $\hat{\mathbf{x}}$ is polynomial in m. □

Acknowledgments. We would like to thank Kang Yang for his helpful comments on the related work of group signatures, and the anonymous reviewers of PKC 2015 for kind suggestions on our paper. We are also grateful to Sherman Chow for bringing their work to our attention.

References

1. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer-Verlag, 2010.

2. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer Berlin / Heidelberg, 2010.
3. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer Berlin / Heidelberg, 2010.
4. S. Agrawal, D. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In D. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer Berlin Heidelberg, 2011.
5. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 99–108, New York, NY, USA, 1996. ACM.
6. M. Ajtai. Generating hard instances of the short basis problem. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *ICALP 1999*, volume 1644 of *LNCS*, pages 706–706. Springer Berlin / Heidelberg, 1999.
7. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *In STACS*, pages 75–86, 2009.
8. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer Berlin Heidelberg, 2009.
9. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure group signature scheme. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer-Verlag, 2000.
10. G. Ateniese, D. Song, and G. Tsudik. Quasi-efficient revocation of group signatures. In M. Blaze, editor, *Financial Cryptography*, volume 2357 of *Lecture Notes in Computer Science*, pages 183–197. Springer Berlin Heidelberg, 2002.
11. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer Berlin Heidelberg, 2003.
12. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *13th ACM Conference on Computer and Communications Security (CCS)*, pages 390–399, New York, NY, USA, 2006. ACM.
13. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM conference on Computer and Communications Security (CCS)*, pages 62–73. ACM Press, 1993.
14. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In ASIACRYPT 2014 (to appear).
15. D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer-Verlag, 2004.
16. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag, 2004.
17. D. Boneh, V. Nikolaenko, and G. Segev. Attribute-based encryption for arithmetic circuits. Cryptology ePrint Archive, Report 2013/669, 2013.
18. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *11th ACM Conference on Computer and Communications Security (CCS)*, pages 168–177, New York, NY, USA, 2004. ACM.

19. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In P. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517. Springer Berlin / Heidelberg, 2010.
20. X. Boyen and B. Waters. Compact group signatures without random oracles. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 427–444. Springer-Verlag, 2006.
21. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In T. Okamoto and X. Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer-Verlag, 2007.
22. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Innovations in Theoretical Computer Science, ITCS*, pages 309–325, 2012.
23. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 97 –106, 2011.
24. Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *5th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 1–12, New York, NY, USA, 2014. ACM.
25. E. Bresson and J. Stern. Efficient revocation in group signatures. In K. Kim, editor, *PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 190–206. Springer Berlin Heidelberg, 2001.
26. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *11th ACM Conference on Computer and Communications Security (CCS)*, pages 132–145. ACM Press, 2004.
27. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer Berlin Heidelberg, 2002.
28. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer-Verlag, 2004.
29. J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In I. Visconti and R. Prisco, editors, *Security and Cryptography for Networks (SCN)*, volume 7485 of *LNCS*, pages 57–75. Springer Berlin Heidelberg, 2012.
30. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer Berlin / Heidelberg, 2004.
31. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer Berlin / Heidelberg, 2010.
32. D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
33. L. Chen and J. Li. Flexible and scalable digital signatures in TPM 2.0. In *20th ACM Conference on Computer and Communications Security (CCS)*, pages 37–48. ACM Press, 2013.
34. Y. Dodis, O. Rafail, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38:97–139, 2008.
35. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. Odlyzko, editor, *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer Berlin Heidelberg, 1987.
36. C. Gentry, S. Gorbunov, S. Halevi, V. Vaikuntanathan, and D. Vinayagamurthy. How to compress (reusable) garbled circuits. Cryptology ePrint Archive, Report 2013/687, 2013.
37. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 197–206, New York, NY, USA, 2008. ACM.

38. S. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 395–412. Springer Berlin / Heidelberg, 2010.
39. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer-Verlag, 2006.
40. J. Groth. Fully anonymous group signatures without random oracles. In K. Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 164–180. Springer-Verlag, 2007.
41. T. C. Group. TCG TPM specification 1.2. Available at `http://www.trustedcomputinggroup.org`, 2003.
42. T. C. Group. TCG TPM specification 2.0. Available at `http://www.trustedcomputinggroup.org/resources/tpm_library_specification`, 2013.
43. I. P. W. Group, VSC Project. Dedicated short range communications (DSRC), 2003.
44. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 41–61. Springer Berlin Heidelberg, 2013.
45. A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 345–361. Springer Berlin Heidelberg, 2014.
46. B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 571–589. Springer-Verlag, 2012.
47. B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 609–627. Springer-Verlag, 2012.
48. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the isis problem, and applications. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer Berlin Heidelberg, 2013.
49. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: simpler, tighter, shorter, ring-based. In *PKC 2015 (to appear)*.
50. V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In R. Cramer, editor, *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer Berlin Heidelberg, 2008.
51. V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer Berlin Heidelberg, 2012.
52. D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer Berlin Heidelberg, 2011.
53. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer Berlin Heidelberg, 2012.
54. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37:267–302, 2007.
55. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 282–298. Springer Berlin Heidelberg, 2003.
56. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 333–342, New York, NY, USA, 2009. ACM.

57. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 145–166. Springer Berlin / Heidelberg, 2006.
58. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37 Annual ACM Symposium on Theory of Computing (STOC)*, pages 84–93, New York, NY, USA, 2005. ACM.
59. J. Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6):1757–1768, Nov 1996.

A The Proof of π_2 in Our Group Signature

In this section, the notations are the same as in the description of our group signature in Section 5. Let $\mathbf{t}_0 := \mathbf{A}_1\mathbf{c} + \mathbf{A}_{2,1}\mathbf{x}_2, \mathbf{t}_1 := \mathbf{A}_1\mathbf{c}$. Let $\mathbf{y}_0 = \mathbf{e}, \mathbf{y}_1 = \mathbf{x}_1 \in \mathbb{Z}^m$, and $\mathbf{y}_2 = \mathbf{v}_j \in \mathbb{Z}^\ell$. In our group signature, we need to generate a NIZKPoK proof (*i.e.*, π_2) for the following relation:

$$R_{com} = \{(\mathbf{A}_1, \mathbf{D}, \mathbf{t}_0, \mathbf{t}_1, \eta; \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2) \in \mathbb{Z}_q^{n\times(m+l)} \times \mathbb{Z}_q^{2n} \times \mathbb{R} \times \mathbb{Z}_q^{2m} \times \mathbb{Z}^\ell : \\ \mathbf{t}_0 = (p\mathbf{A}_1)\mathbf{y}_0 - \mathbf{D}\mathbf{y}_2, \mathbf{t}_1 = (p\mathbf{A}_1)\mathbf{y}_0 + \mathbf{A}_1\mathbf{y}_1, \text{ and } \|\mathbf{y}_i\| \le \eta\}.$$

The proof is essentially achieved by combining our protocol from Section 4.2 (*i.e.*, for the equation $\mathbf{t}_0 = (p\mathbf{A}_1)\mathbf{y}_0 - \mathbf{D}\mathbf{y}_2$) and the standard protocol for R_{ISIS} from [44,51] (*i.e.*, for the equation $\mathbf{t}_1 = (p\mathbf{A}_1)\mathbf{y}_0 + \mathbf{A}_1\mathbf{y}_1$). We present the basic combined protocol with single-bit challenge in Fig. 3.

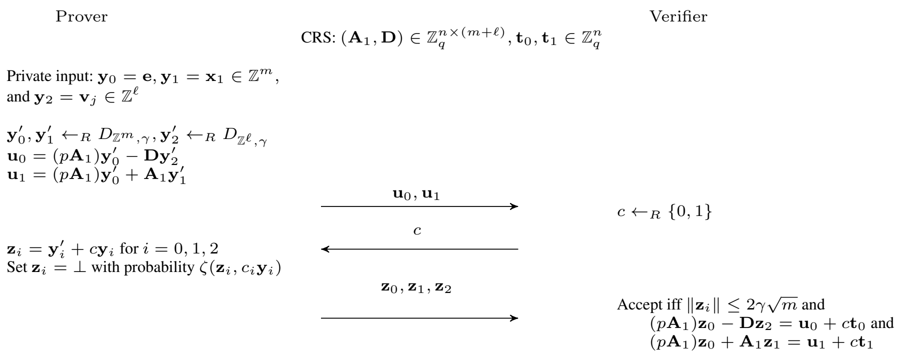
Prover
Verifier
CRS: $(\mathbf{A}_1, \mathbf{D}) \in \mathbb{Z}_q^{n\times(m+\ell)}, \mathbf{t}_0, \mathbf{t}_1 \in \mathbb{Z}_q^n$
Private input: $\mathbf{y}_0 = \mathbf{e}, \mathbf{y}_1 = \mathbf{x}_1 \in \mathbb{Z}^m$, and $\mathbf{y}_2 = \mathbf{v}_j \in \mathbb{Z}^\ell$
$\mathbf{y}'_0, \mathbf{y}'_1 \leftarrow_R D_{\mathbb{Z}^m,\gamma}, \mathbf{y}'_2 \leftarrow_R D_{\mathbb{Z}^\ell,\gamma}$
$\mathbf{u}_0 = (p\mathbf{A}_1)\mathbf{y}'_0 - \mathbf{D}\mathbf{y}'_2$
$\mathbf{u}_1 = (p\mathbf{A}_1)\mathbf{y}'_0 + \mathbf{A}_1\mathbf{y}'_1$
$\mathbf{u}_0, \mathbf{u}_1$
$c \leftarrow_R \{0,1\}$
c
$\mathbf{z}_i = \mathbf{y}'_i + c\mathbf{y}_i$ for $i = 0, 1, 2$
Set $\mathbf{z}_i = \perp$ with probability $\zeta(\mathbf{z}_i, c_i\mathbf{y}_i)$
$\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2$
Accept iff $\|\mathbf{z}_i\| \le 2\gamma\sqrt{m}$ and
$(p\mathbf{A}_1)\mathbf{z}_0 - \mathbf{D}\mathbf{z}_2 = \mathbf{u}_0 + c\mathbf{t}_0$ and
$(p\mathbf{A}_1)\mathbf{z}_0 + \mathbf{A}_1\mathbf{z}_1 = \mathbf{u}_1 + c\mathbf{t}_1$

Fig. 3. The basic Σ-protocol for Relation R_{com}

As before, due to the use of a one-bit challenge, given two accepted transcripts with the same first message but different challenges: $((\mathbf{u}_0, \mathbf{u}_1), 1, (\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2))$ and $((\mathbf{u}_0, \mathbf{u}_1), 0, (\mathbf{z}'_0, \mathbf{z}'_1, \mathbf{z}'_2))$, one can recover a "weak" witness $(\hat{\mathbf{z}}_0, \hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2) \in \mathbb{Z}^{2m+\ell}$ satisfying $\mathbf{t}_0 = (p\mathbf{A}_1)\hat{\mathbf{z}}_0 - \mathbf{D}\hat{\mathbf{z}}_2, \mathbf{t}_1 = (p\mathbf{A}_1)\hat{\mathbf{z}}_0 + \mathbf{A}_1\hat{\mathbf{z}}_1$, and $\|\hat{\mathbf{z}}_i\| \le 4\gamma\sqrt{m}$, where $\hat{\mathbf{z}}_i = \mathbf{z}_i - \mathbf{z}'_i$ and $\gamma = \eta m^{1.5}$. Besides, the simulator for the special Honest-Verifier Zero Knowledge (HVZK) property simply works as follows: given a challenge $c \in \{0, 1\}$, it chooses

$\mathbf{z}_0, \mathbf{z}_1 \leftarrow_R D_{\mathbb{Z}^m,\gamma}, \mathbf{z}_2 \leftarrow_R D_{\mathbb{Z}^\ell,\gamma}$, and computes $\mathbf{u}_0 = (p\mathbf{A}_1)\mathbf{z}_0 - \mathbf{D}\mathbf{z}_2 - c\mathbf{t}_0$, and $\mathbf{u}_1 = (p\mathbf{A}_1)\mathbf{z}_0 + \mathbf{A}_1\mathbf{z}_1 - c\mathbf{t}_1$. Then, it sets $\mathbf{z}_i = \perp$ with probability $1 - \frac{1}{M_l}$, and outputs $((\mathbf{u}_0, \mathbf{u}_1), c, (\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2))$. By Lemma 3, the distribution of $(\mathbf{u}_0, \mathbf{u}_1)$ is statistically close to uniform over $\mathbb{Z}_q^n \times \mathbb{Z}_q^n$ (since $(p\mathbf{A}_1)\mathbf{z}_0$ and $\mathbf{A}_1\mathbf{z}_1$ are both statistically close to $\mathbb{Z}_q^n$), which is the same as that in the transcript of a real proof. In addition, by [51, Th. 4.6], the distribution of $(\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2)$ is also statistically close to that in the real transcript.

Like the protocol in [51,44] and ours in Section 4.2, we have to use a $t = \omega(\log n)$ parallel repetitions of the basic protocol in Fig. 3 to achieve negligible soundness error. The final NIZKPoK protocol used in our group signature is obtained by using the Fiat-Shamir transformation in a standard way. Namely, the challenges $\mathbf{c} = (c_1, \dots, c_t)$ are computed by using a hash function (modeled as a random oracle) with inputs all the public information such as all the t independent pairs $(\mathbf{u}_0, \mathbf{u}_1)$, and the message M to be signed.

B One-time Strongly Unforgeable Signatures

In this section, we recall the definition and security model of signatures. Formally, a signature scheme $\mathcal{SIG}$ consists of three PPT algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$:

- $\mathsf{KeyGen}(1^n)$: Take as input the security parameter n, output a verification key svk and a signing key ssk.
- $\mathsf{Sign}(ssk, M)$: Take as inputs a signing key ssk and a message M (in some implicit message space), output a signature σ.
- $\mathsf{Ver}(svk, M, \sigma)$: Take as inputs a verification key svk, a message M, and a signature σ, output a bit $b \in \{0, 1\}$ (where $b = 1$ indicates "acceptance" and $b = 0$ indicates "rejection").

We require that for all $(svk, ssk) \leftarrow \mathsf{KeyGen}(n)$, and all M in the message space, the probability that $\mathsf{Ver}(svk, M, \mathsf{Sign}(ssk, M)) = 1$ holds with overwhelming probability. In particular, σ is said to be a valid signature for the message M under the verification key svk if $\mathsf{Ver}(svk, M, \sigma) = 1$.

A signature scheme is said to be strongly existentially unforgeable under a one-time chosen message attack if for any PPT forger $\mathcal{F}$, its advantage

$$\mathrm{Adv}^{\text{ot-sEUF}}_{\mathcal{SIG},\mathcal{F}}(n) = \Pr[\mathbf{Exp}^{\text{ot-sEUF}}_{\mathcal{SIG},\mathcal{F}}(n) = 1]$$

in the following experiment is negligible in the security parameter n:

```
$\mathbf{Experiment}\ \mathbf{Exp}^{\text{ot-sEUF}}_{\mathcal{SIG},\mathcal{F}}(n)$
    $(svk, ssk) \leftarrow \mathsf{KeyGen}(1^n)$
    $(st, M^*) \leftarrow \mathcal{F}(svk)$
    $\sigma^* = \mathsf{Sign}(ssk, M^*)$
    $(M', \sigma') \leftarrow \mathcal{F}(st, \sigma^*)$
    Return 1 if
        $\mathsf{Ver}(svk, M', \sigma') = 1$ and $(M', \sigma') \neq (M^*, \sigma^*)$
    Else return 0
```