Floppy-Sized Group Signatures from Lattices

Cecilia Boschini[1,2](✉), Jan Camenisch[1], and Gregory Neven[1]

[1] IBM Research, Zurich, Switzerland
[2] Università della Svizzera Italiana, Lugano, Switzerland
{bos,jca,nev}@zurich.ibm.com

Abstract. We present the first lattice-based group signature scheme whose cryptographic artifacts are of size small enough to be usable in practice: for a group of 2^{25} users, signatures take 910 kB and public keys are 501 kB. Our scheme builds upon two recently proposed lattice-based primitives: the verifiable encryption scheme by Lyubashevsky and Neven (Eurocrypt 2017) and the signature scheme by Boschini, Camenisch, and Neven (IACR ePrint 2017). To achieve such short signatures and keys, we first re-define verifiable encryption to allow one to encrypt a function of the witness, rather than the full witness. This definition enables more efficient realizations of verifiable encryption and is of independent interest. Second, to minimize the size of the signatures and public keys of our group signature scheme, we revisit the proof of knowledge of a signature and the proofs in the verifiable encryption scheme provided in the respective papers.

Keywords: Lattices · Group Signature · Verifiable Encryption

1 Introduction

Lattice-based cryptography has made substantial advances and now includes public-key encryption schemes [33, 34] and digital signature schemes [17, 18, 30] that are essentially as practical as those based on traditional number-theoretic assumptions: all keys and outputs are less than 1 kB for 128 bits of security. Somewhat more complex primitives such as identity-based encryption [22, 18] can be implemented with keys and ciphertexts being around 4 kB, and the best blind signature scheme [38] has artifacts of around 100 kB. For group signatures [16], however, the lattice-based schemes known are much less efficient than their traditional counterparts, despite the attention they have recently received.

In a group signature scheme, the group manager provides distinct secret keys to each user, who is then able to sign messages anonymously on behalf of the group. While anyone can check that a message was signed by a group member, only the opener is able to recover the identity of the originator of a signature. Group signatures are particularly useful in scenarios where remote devices need to be authenticated as valid devices, but privacy imposes that individual devices can only be identified by a designated authority. Examples include government-issued electronic identity (eID) cards, where each issued smart card creates identity claims as signed statements about its attributes, without needing to fully identify its owner [9], or remote anonymous attestation of computing platforms, where devices prove which software they execute [12].

A typical approach to construct a group signature scheme is to use a signature scheme, an encryption scheme, and a non-interactive zero-knowledge proof of knowledge (NIZK PoK) [3, 15, 8] as follows. The group public key consists of the group manager's signature public key and the opener's encryption public key. A user's secret key is a signature by the group manager on the identity of the user. To sign a message, the user encrypts her identity under the opener's public key and creates a NIZK PoK of a signature on the encrypted value.

The main obstacle in achieving an efficient scheme with this approach is the efficiency of the NIZK PoK and the choice of signature and encryption schemes that allow for an efficient NIZK PoK. In this paper, we build a dynamic group signature scheme by combining the recent signature scheme with protocols by Boschini, Camenisch, and Neven [11] and the recent (verifiable) encryption scheme by Lyubashevsky and Neven [32]. Both these schemes already come with NIZK proofs of knowledge of a signature and of a plaintext, but their straightforward combination results in a group signature scheme that is not practical due to its large signature size.

Our Techniques and Results. Boschini et al. [11] presented a (relaxed) signature scheme allowing for efficient zero-knowledge proofs of knowledge of a signature on a hidden message, where a signature on a polynomial with small coefficients $\mathbf{m}$ is a vector $\mathbf{S}$ of small-coefficient polynomials (or "short" vector) such that $[\mathbf{A}|\mathbf{B}|\mathbf{C}+\mathbf{m}\mathbf{G}|\mathbf{1}]\mathbf{S} = \mathbf{u}$, where the public key contains row vectors $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{G}$ and a polynomial $\mathbf{u}$. To prove knowledge of a signature on a hidden message, the prover first generates a commitment $\mathbf{F} = \mathbf{b}^{-1}(\mathbf{C} + \mathbf{m}\mathbf{G} + \mathbf{E})$ to $\mathbf{m}$, where $\mathbf{b}$ is a random small-coefficient polynomial and $\mathbf{E}$ is an error vector. The commitment $\mathbf{F}$ can be plugged into the verification equation by computing a short vector $\mathbf{S}'$ such that $[\mathbf{A}|\mathbf{B}|\mathbf{F}|\mathbf{1}]\mathbf{S}' = \mathbf{u}$. The prover can then use Lyubashevsky's Fiat-Shamir with aborts technique [30] to prove knowledge of

$$\text{(I) } [\mathbf{A}|\mathbf{B}|\mathbf{F}|\mathbf{1}]\bar{\mathbf{S}} = \bar{\mathbf{c}}\mathbf{u} \qquad \text{(II) } [\mathbf{F}^T|\mathbf{G}^T|\mathbf{1}] \begin{bmatrix} \bar{\mathbf{b}} \\ \bar{\mathbf{m}} \\ \bar{\mathbf{E}} \end{bmatrix} = \bar{\mathbf{c}}'\mathbf{C} \ .$$

The relaxed verifiable encryption scheme of Lyubashevsky and Neven [32] can encrypt a witness $\mathbf{x}$ to a relation $\mathbf{M}\mathbf{x} = \mathbf{y}$ so that decryption is guaranteed to yield $(\bar{\mathbf{x}}, \bar{\mathbf{c}})$ such that $\mathbf{M}\bar{\mathbf{x}} = \bar{\mathbf{c}}\mathbf{y}$. The most straightforward way to build a group signature scheme would be to combine it with the above building blocks, letting a user's signing key be given by a signature $\mathbf{S}$ by the group manager on the user's identity $\mathbf{m}$, and letting a group signature be a non-interactive proof of relations (I) and (II), combined with a verifiable encryption to allow the opener to recover the user's identity $\mathbf{m}$.

The problem with this approach is that the Lyubashevsky-Neven verifiable encryption scheme encrypts the full witness $[\mathbf{S} \ ; \ \bar{\mathbf{b}} \ ; \ \bar{\mathbf{m}} \ ; \ \bar{\mathbf{E}}^T]$, rather than just the witness $\mathbf{m}$, resulting in a very long signature size. In this paper, we define a variant of relaxed verifiable encryption that encrypts only part of the witness, resulting in a much shorter signature size. In this way, given $\mathbf{F}$ as before, it is possible to encrypt the message $\mathbf{m}$ and still prove that it was used to construct $\mathbf{F}$, without having to also encrypt $\mathbf{S}$, $\mathbf{b}$, and $\mathbf{E}$. Moreover, we prove relations (I) and (II) in two separate proofs, resulting in better parameters.

Our group signature scheme satisfies anonymity and traceability as defined by Bellare et al. [7] in the random-oracle model. Analogously to the non-lattice-based world, where schemes under weak assumptions do exist [6, 7] but truly practical schemes typically require stronger assumptions [3, 10], we also prove our scheme secure under relatively strong assumptions. Namely, we follow the approach by Boschini et al. [11] and use two interactive assumptions that can be interpreted in two different ways. One can either believe the interactive assumptions as stated, in which case we obtain a tight security reduction and the most efficient parameters for our scheme, resulting in signatures of 910 kB for a group of 2^{25} users and 80 bits of security. Alternatively, one can see our assumptions as being implied by the standard Ring-SIS and Ring-LWE assumptions through a complexity leveraging argument. To compensate for the loose reduction, the parameters increase, resulting in signatures of 1.72 MB.

Related Work. The early lattice-based group signature schemes [23, 13] have signature sizes that are linear in the number of group members and are therefore mainly proofs of concepts, unsuitable for any practical application. Later schemes [24, 28, 37] are asymptotically more efficient with signature sizes being logarithmic in the number of users.

Making use of the advances in lattice-based signature schemes, a number of group signature schemes were proposed following the general construction approach we have outlined earlier [24, 26, 27, 28, 29, 40]. These schemes use as proof of knowledge protocols either an adaptation of Stern's protocol [39] or the "single-bit-challenge" version of the lattice-based Fiat-Shamir protocol by Lyubashevsky [30]. As these proofs have soundness error 2/3 and 1/2, respectively, they need to be repeated sufficiently many times in parallel, resulting in group signature schemes that can hardly be considered practical. None of these scheme give concrete parameters, providing asymptotic efficiency analyses instead. The only exception is the scheme by Libert et al. [26] which is the most efficient scheme prior to ours, with signatures over 60 MB and public keys of 4.9 MB for a group size of only 2^{10} users for 80 bits of security – still much less efficient than ours.

2 Prerequisites

We denote vectors and matrices with upper-case letters. Column vectors are denoted as $V = \begin{bmatrix} v_1 \; ; \ldots ; \; v_n \end{bmatrix}$ and row vectors as $V = \begin{bmatrix} v_1 \ldots v_n \end{bmatrix}$. Sampling and element x from a distribution $\mathcal{D}$ will be denoted as $x \xleftarrow{\$} \mathcal{D}$. If x is sampled from a uniform over a set A, we will abuse the notation and write $x \xleftarrow{\$} A$. With $x \leftarrow a$ we will denote that x is assigned the value a. When necessary, we will denote the uniform distribution over a set S as $U(S)$.

2.1 Polynomial Rings

Consider the polynomial ring $\mathcal{R}_q = \mathbb{Z}_q/\langle \mathbf{x}^n + 1 \rangle$ for a prime $q \equiv 5 \bmod 8$. Elements in the ring are polynomials of degree at most $n-1$ with coefficients in $[-(q-1)/2, (q-1)/2]$ and operations between ring elements are done modulo q. Let $\deg(\mathbf{a})$ be the degree of the polynomial $\mathbf{a}$. For an element $\mathbf{a} = \sum_{i=0}^{n-1} a_i \mathbf{x}^i$

in $\mathcal{R}_q$, the standard norms are computed as $\|\mathbf{a}\|_1 = \sum_i |a_i|$, $\|\mathbf{a}\| = \sqrt{\sum_i a_i^2}$ and $\|\mathbf{a}\|_\infty = \max |a_i|$. For any $K|n$, we can construct a subring $\mathcal{R}_q^{(K)}$ of $\mathcal{R}_q$ as the subset of elements $\mathbf{a} \in \mathcal{R}_q$ such that $\mathbf{a} = \sum_{i=0}^{K-1} a_i \mathbf{x}^{in/K}$. For integer p, $\mathcal{R}_p$ (resp., $\mathcal{R}_p^{(K)}$) is the subset of $\mathcal{R}_q$ (resp., $\mathcal{R}_q^{(K)}$) that contains polynomials with coefficients in $[-(p-1)/2, (p-1)/2]$. Lemma 1 shows that the ring $\mathcal{R}_q$ has a large set of invertible elements that are easy to identify.

Lemma 1 ([32, Lemma 2.2]). *Let $\mathcal{R}_q = \mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ where $n > 1$ is a power of 2 and q is a prime congruent to 5 mod 8. This ring has exactly $2q^{n/2} - 1$ elements without an inverse. Moreover, every non-zero polynomial $\mathbf{a}$ in $\mathcal{R}_q$ with $\|\mathbf{a}\|_\infty < \sqrt{q/2}$ has an inverse.*

There are some easy bounds on the norm of the product of polynomials.

Lemma 2. *For $\mathbf{a}, \mathbf{b} \in \mathcal{R}_q$ it holds: $\|\mathbf{ab}\|_\infty \le \min\{\|\mathbf{a}\|_\infty \|\mathbf{b}\|_1, (q-1)/2\}$. Moreover, let $\mathbf{a}$, $\mathbf{b} \in \mathcal{R}_q$ be such that $n\|\mathbf{a}\|_\infty \cdot \|\mathbf{b}\|_\infty \le (q-1)/2$. Then we have that $\|\mathbf{ab}\| \le \|\mathbf{a}\|\|\mathbf{b}\|\sqrt{n}$ and $\|\mathbf{ab}\|_\infty \le \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty n \le \frac{q-1}{2}$.*

2.2 Lattices

An integer lattice is an additive subgroup of $\mathbb{Z}^n$. Every lattice Λ is generated by a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \in \mathbb{Z}^{n \times m}$, where m is called *dimension* of the lattice. Such lattice is denoted by $\Lambda = \mathcal{L}(\mathbf{B})$. If $k = n$ and the vectors in the basis are linearly independent the lattice is a *full-rank* lattice. The Gram-Schmidt orthogonalization of a full-rank basis $\mathbf{B}$ is denoted by $\tilde{\mathbf{B}} = \{\tilde{\mathbf{b}}_1, \ldots \tilde{\mathbf{b}}_n\}$. Let $\tilde{\lambda}(\mathcal{L}(\mathbf{B})) = \min_{\mathbf{B}' \text{ s.t. } \mathcal{L}(\mathbf{B}')=\mathcal{L}(\mathbf{B})} \|\tilde{\mathbf{B}}'\|$. For a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, $\Lambda^\perp$ is the lattice: $\Lambda^\perp = \mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \,|\, \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\} \subseteq \mathbb{Z}^m$. We define the *discrete Gaussian distribution* centered in $\mathbf{c}$ with standard deviation σ on a full-rank lattice Λ as $\mathcal{D}_{\Lambda,\mathbf{c},\sigma}(\mathbf{v}) = e^{-\frac{\pi\|\mathbf{v}-\mathbf{c}\|^2}{\sigma^2}} / \sum_{\mathbf{u}\in\Lambda} e^{-\frac{\pi\|\mathbf{u}-\mathbf{c}\|^2}{\sigma^2}}$ for all $\mathbf{v} \in \Lambda$, and 0 on all the other points in the space. Let $\mathcal{D}^\perp_{\mathbf{A},\mathbf{u},\sigma}$ be the distribution of the vectors $\mathbf{s}$ such that $\mathbf{s} \sim \mathcal{D}_{\mathbb{Z}^n,\mathbf{0},\sigma}$ conditioned on $\mathbf{A}\mathbf{s} = \mathbf{u} \bmod q$.

Lemma 3 (cf. [5, Lemma 1.5], [30, Lemma 4.4]). *Let $\mathbf{A} \in \mathbb{Z}^{n\times m}$ with $2^{11} < m$ and $\mathbf{u} \in \mathbb{Z}_q^n$. For $\sigma \ge \tilde{\lambda}(\mathcal{L}^\perp(\mathbf{A}))$ it holds:*
$\mathsf{Pr}_{\mathbf{s} \overset{\$}{\leftarrow} \mathcal{D}^\perp_{\mathbf{A},\mathbf{u},\sigma}}(\|\mathbf{s}\| > 1.05\sigma\sqrt{m}) < 2^{-5}$ *and* $\mathsf{Pr}_{\mathbf{s} \overset{\$}{\leftarrow} \mathcal{D}^\perp_{\mathbf{A},\mathbf{u},\sigma}}(\|\mathbf{s}\|_\infty > 8\sigma) < m2^{-46}$.
In particular, the inequalities hold also when $\mathbf{s} \overset{\$}{\leftarrow} \mathcal{D}_{\mathbb{Z}^m,\mathbf{u},\sigma}$.

2.3 Lattices over Rings

Lattices over the polynomial ring $\mathcal{R}_q$ can be defined similarly to lattices over $\mathbb{Z}_q$. Indeed, given $\mathbf{A} \in \mathcal{R}_q^{1\times m}$ we can construct m-dimensional lattice $\mathcal{L}^\perp(\mathbf{A})$ as $\Lambda^\perp = \mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{V} \in (\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle)^m \,|\, \mathbf{A}\mathbf{V} = \mathbf{0} \bmod q\} \subseteq \mathcal{R}_q^m$. Consider the obvious embedding that maps a polynomial to the vector of its coefficients. Then $\Lambda^\perp$ can be also seen as a nm-dimensional integer lattice over $\mathbb{Z}$. With a slight abuse of notation, we will write $\mathbf{y} \overset{\$}{\leftarrow} \mathcal{D}_{\mathcal{R}_q,\mathbf{u},\sigma}$ to indicate that $\mathbf{y}$ was sampled from $\mathcal{D}_{\mathbb{Z}^n,\mathbf{u},\sigma}$ and then mapped to $\mathcal{R}_q$. Similarly, we omit the $\mathbf{0}$ and write $\left[\mathbf{y}_1 \ldots \mathbf{y}_k\right] \overset{\$}{\leftarrow} \mathcal{D}^k_{\mathcal{R}_q,\sigma}$ to mean that a vector $\mathbf{y}$ is generated according to

$\mathcal{D}_{\mathbb{Z}^{kn},\mathbf{0},\sigma}$ and then gets interpreted as k polynomials $\mathbf{y}_i$.
We recall some results about sampling an element from a Gaussian distribution over a lattice given some trapdoor.

Theorem 1 (adapted from [35]). *Let* $\mathbf{A}$ *be a vector in* $\mathcal{R}_q^{1\times\ell}$ *and* $\mathbf{X}$ *be a matrix in* $\mathcal{R}_q^{\ell\times m}$*. Also define the* gadget matrix $\mathbf{G} = \begin{bmatrix}1 \lceil q^{1/m}\rceil \dots \lceil q^{(m-1)/m}\rceil\end{bmatrix}$*. Then for any invertible* $\mathbf{m} \in \mathcal{R}_q$*, there is an algorithm that can sample from the distribution* $\mathcal{D}^{\perp}_{[\mathbf{A}\ \mathbf{AX}+\mathbf{mG}],\mathbf{u},\sigma}$ *for any* $\sigma \sim q^{\frac{1}{m}} s_1(\mathbf{X}) > \tilde{\lambda}(\Lambda^{\perp}([\mathbf{A}\ \mathbf{AX}+\mathbf{mG}]))$ *for any* $\mathbf{u} \in \mathcal{R}_q$*.*

Lemma 4. *Suppose* $\mathbf{U} \in \mathcal{R}_q^{1\times k}$ *and* $\mathbf{V} \in \mathcal{R}_q^{1\times m}$ *are polynomial vectors, and* $\mathbf{B}_U, \mathbf{B}_{(U,V)}$ *are bases of* $\Lambda^{\perp}(\mathbf{U})$ *and* $\Lambda^{\perp}([\mathbf{U}\ \mathbf{V}])$ *respectively such that* $\|\tilde{\mathbf{B}}_U\|$, $\|\tilde{\mathbf{B}}_{(U,V)}\| < \sigma\sqrt{\pi/\ln(2n+4)}$*. Then, there exists an algorithm* $\mathsf{SampleD}(\mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{u}, \sigma)$*, where* $\mathbf{B}$ *is either* $\mathbf{B}_U$ *or* $\mathbf{B}_{(U,V)}$*, that can efficiently sample from the distribution* $D^{\perp}_{[\mathbf{U}\ \mathbf{V}],\mathbf{u},\sigma}$ *for any* $\mathbf{u} \in \mathcal{R}_q$*.*

2.4 Hardness Assumptions

We recall two well-studied lattice problems over rings: Ring-SIS and Ring-LWE.

Definition 1. (Ring-SIS$_{m,q,\beta}$ problem) *The Ring-SIS*$_{m,q,\beta}$ *problem is given a vector* $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$ *to find a vector* $\mathbf{S} \in \mathcal{R}_q^{m}$ *such that* $[\mathbf{A}\ \mathbf{1}]\,\mathbf{S} = \mathbf{0}$ *and* $\|\mathbf{S}\| \leq \beta$*.*

Definition 2. *The* Ring-LWE$_D$ distribution *outputs pairs* $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}_q \times \mathcal{R}_q$ *such that* $\mathbf{b} = \mathbf{as} + \mathbf{e}$ *for a uniformly random* $\mathbf{a}$ *from* $\mathcal{R}_q$ *and* $\mathbf{s}, \mathbf{e}$ *sampled from distribution* D*. The* Ring-LWE$_{k,D}$ decisional problem *on ring* $\mathcal{R}_q$ *with distribution* D *is to distinguish whether* k *pairs* $(\mathbf{a}_1, \mathbf{b}_1), \dots, (\mathbf{a}_k, \mathbf{b}_k)$ *were sampled from the Ring-LWE*$_D$ *distribution or from the uniform distribution over* $\mathcal{R}_q^2$*.*

There is a polynomial-time reduction from solving the shortest vector problem over rings to Ring-SIS [31, Theorem 5.1] and a polynomial-time quantum reduction from solving the shortest vector problem over rings to Ring-LWE with Gaussian error distribution (cf. [33]). The root Hermite factor δ introduced by Gama and Nguyen [21] is used to estimate the hardness of the lattice problems for given parameters in the security reductions.

Boschini et al. [11] introduce new hardness assumptions to be able to prove their schemes secure with or without complexity leveraging. The idea is to state the assumptions in two forms, selective and adaptive. The schemes are proved secure assuming the adaptive variants of the assumptions. Then, a reduction from adaptive to selective is proved using complexity leveraging, and Ring-SIS and Ring-LWE are reduced to the selective version. Hence, allowing the use of complexity leveraging it is possible to base the security of the schemes on Ring-SIS and Ring-LWE, otherwise security is guaranteed under the adaptive version of the new hardness assumptions (cf. Assumptions 1 and 3).

Assumption 1 *Consider the following game between an adversary* A *and a challenger for fixed* $m \in \mathbb{N}$ *and distribution* D*:*

1. *The challenger outputs a uniformly random* $\mathbf{C} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ *to* A.
2. A *sends back* $\mathbf{m} \in \mathcal{U}$.
3. *The challenger picks a uniformly random bit* $b \xleftarrow{\$} \{0,1\}$. *If* $b = 1$, *it samples an error vector* $\mathbf{E} \xleftarrow{\$} D^m$ *and* $\mathbf{s} \xleftarrow{\$} D$, *and sends* $\mathbf{F} = (\mathbf{C} + \mathbf{m}\mathbf{G} - \mathbf{E})\mathbf{s}^{-1}$ *to* A. *Otherwise, it sends a uniform* $\mathbf{F} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ *to* A.
4. A *sends a bit* b' *to the challenger.*

The advantage of A *in winning the game is* $\left|\mathsf{Pr}(b = b') - \frac{1}{2}\right|$. *The assumption states that no PPT* A *can win the previous game with non-negligible advantage.*

Assumption 2 (Selective variant of Assumption 1.) *Consider the game of Assumption 1, but with steps 1 and 2 switched, meaning,* A *outputs* $\mathbf{m} \in \mathcal{U}$ *before being given* $\mathbf{C}$. *The assumption states that no PPT adversary can win this previous game with non-negligible advantage.*

Boschini et al. proved that Assumption 2 is at least as hard as Ring-LWE with m samples and distribution D. It is possible to reduce Assumption 2 to 1 with a complexity leveraging argument by guessing the value of $\mathbf{m} \in \mathcal{U}$.

Assumption 3 *Let* $\bar{\Sigma} = \{(\mathbf{c}_1, \mathbf{S}, \mathbf{c}_2) \in \bar{\mathcal{C}} \times \mathcal{R}_q^{3+2m} \times \mathcal{R}_q : \ \|\mathbf{S}\| \leq N' \ \wedge \ \|\mathbf{c}_2\| \leq C'\}$ *for some fixed parameters. Consider the following game between an adversary* A *and a challenger for fixed* $m \in \mathbb{N}$ *and distribution* D:

1. *The challenger chooses* $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$, $\mathbf{C} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$, *and* $\mathbf{X} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$. *It sets* $\mathbf{A} = [\mathbf{a}|\mathbf{1}]$ *and* $\mathbf{B} = \mathbf{A}\mathbf{X} + \mathbf{G}$, *where* $\mathbf{G} = \left[1\ \lceil q^{1/m}\rceil\ \ldots\ \lceil q^{(m-1)/m}\rceil\right]$.
2. *The challenger runs* A *on input* $\left[\mathbf{A}\ \mathbf{B}\ \mathbf{C}\ \mathbf{1}\right]$, *giving it access to a random oracle* $\mathcal{H} : \{0,1\}^* \to \mathcal{R}_q$ *and an oracle* $\mathcal{O}_S$ *that on input* $\mathbf{m} \in \mathcal{U}$ *and a string* $\alpha \in \{0,1\}^*$ *outputs a small vector* $\left[\mathbf{S}\ ;\ \mathbf{0}\right]$ *in the coset* $\mathcal{L}^{\perp}(\left[\mathbf{A}\ \mathbf{B}\ \mathbf{C} + \mathbf{m}\mathbf{G}\ \mathbf{1}\right]) + \mathcal{H}(\alpha)$ *such that* $\|\mathbf{S}\| \leq N_S$.
3. *Algorithm* A *outputs* $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$, $\bar{\alpha} \in \{0,1\}^*$, $\bar{\mathbf{c}}_1 \in \bar{\mathcal{C}}$, *a ring element* $\bar{\mathbf{c}}_2$ *and a vector* $\bar{\mathbf{S}}$. *Algorithm* A *wins the game if* $(\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2) \in \bar{\Sigma}$, $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$, *such that* $\mathbf{S}$ *is a short vector of the coset* $\mathcal{L}^{\perp}(\left[\mathbf{A}\ \mathbf{B}\ \bar{\mathbf{C}}\ \mathbf{1}\right]) + \mathbf{c}_2\mathcal{H}(\bar{\alpha}))$ *where* $\bar{\mathbf{C}} = \bar{\mathbf{c}}_1\mathbf{C} - \bar{\mathbf{m}}\mathbf{G}$, *and* $(\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}, \bar{\alpha})$ *was not queried to the* $\mathcal{O}_S$ *oracle.*

The assumption states that no PPT algorithm A *can win the game with non-negligible probability.*

Assumption 4 (Selective variant of Assumption 3.) *Consider the game of Assumption 3, but where step 1 is preceded with a step where* A, *on input only the security parameter* λ, *outputs the message* $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$, *and in step 3 outputs the remaining items* $\bar{\alpha}$, $\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2 \in \bar{\mathcal{C}}$, *and* $\bar{\mathbf{S}}$. *The assumption states that no PPT adversary can win this previous game with non-negligible advantage.*

Theorem 2 (Hardness of Assumption 4). *Let* A *be a probabilistic algorithm that breaks Assumption 4 in time* t *with probability* ϵ_A. *Then there exists a probabilistic algorithm* B *that either breaks Ring-LWE*$_{m,\mathcal{D}_\sigma}$ *in time* t *with probability* ϵ_A *or Ring-SIS*$_{3+m,q,\beta_s}$ *in time* t *with probability* $\epsilon_B \geq (\epsilon_A - \epsilon_{\mathrm{LWE}})/(2\cdot|\bar{\mathcal{C}}|)$, *where* $\beta_{\mathrm{s}} = N'^2 + \frac{\sigma_t^2}{\pi} n^2(\sqrt{2} + \sqrt{m} + \log n)^2(2\sqrt{2^{K_c}})^2 N'^2 + \frac{\sigma^2}{\pi} n(1 + \sqrt{2} + \log n)^2(C'^2 + (1.05\sigma_t\sqrt{n})^2)$, ϵ_{LWE} *is the probability of breaking the Ring-LWE problem over* $\mathcal{R}_q$ *in time* t, *in the Random Oracle Model.*

The bound β_{s} is different from the original result, as we choose larger message and challenge spaces. From complexity leveraging (guessing $\bar{\mathbf{m}}$ in $\bar{\mathcal{U}}$ and $\bar{c}_1$ in $\bar{\mathcal{C}}$) it follows that breaking Assumption 4 implies breaking Assumption 3.

2.5 Group Signature

A group signature is a set of algorithms $(\mathsf{GPGen}, \mathsf{GKGen}, \mathsf{UKGen}, \mathsf{OKGen}, \mathsf{GSign}, \mathsf{GVerify}, \mathsf{GOpen})$ run by a group manager, an opener and users. The group signature parameters $gpar$ are generated via $\mathsf{GPGen}(1^\lambda)$ (where λ is the security parameter). The group manager and the opener generate their keys running $(gpk, gsk) \leftarrow \mathsf{GKGen}(gpar)$ and $(opk, osk) \leftarrow \mathsf{OKGen}(gpk)$ respectively. If a user wants to join, she sends her identity to the group manager and obtains back her user secret key $usk \leftarrow \mathsf{UKGen}(gsk, id)$. The user can sign a message M on behalf of the group using her secret key with the algorithm $\mathsf{GSign}(usk, gpk, opk, M)$. A signature sig on a message M can be verified with the algorithm $\{1, 0\} \leftarrow \mathsf{GVerify}(M, sig, gpk, opk)$. Finally, the opener can recover the identity of the group member that signed a message M running $id \leftarrow \mathsf{GOpen}(M, sig, osk)$. We require the scheme to be correct (honestly generated signatures satisfy verification and can be opened to the identity of the signer), traceable (the group manager should be able to link every signature to the user who produced it) and anonymous (signatures produced by different users should be indistinguishable).

2.6 One-Time Signature

A One-Time Signature (OTS) scheme for message set $\mathcal{M}$ is a triple $(\mathsf{OTSGen}, \mathsf{OTSSign}, \mathsf{OTSVf})$, where $(sk, vk) \leftarrow \mathsf{OTSGen}(1^\lambda)$ is the key generation algorithm, $ots \leftarrow \mathsf{OTSSign}(sk, msg)$ is the signing algorithm and $0/1 \leftarrow \mathsf{OTSVf}(vk, msg, ots)$ is the verification algorithm. Correctness requires that for all security parameters $\lambda \in \mathbb{N}$ the verification of a honestly generated signature always outputs 1. An OTS is unforgeable if, given sk, vk, no adversary can come up with a signature on a message msg' w.r.t. vk after seeing a signature on msg generated using sk. In particular, the Lamport signature [25] is quantum-secure, thus it can be used with the relaxed Σ-protocol.

2.7 Relaxed ZK proofs

Given two NP-languages $L \subseteq \bar{L}$ defined by the relations $R \subseteq \bar{R}$ respectively, a relaxed Σ-protocol for $L, \bar{L}$ is a three-rounds two-party protocol between PPT algorithms $(\mathcal{P}, \mathcal{V})$ that satisfies standard completeness and zero-knowledge, but where extraction is only guaranteed to output a witness w such that $(x, w) \in \bar{R}$. A protocol can be made non-interactive using Fiat-Shamir transform. Simulation-soundness of the transform can be ensured (cf. [19]) by a property called "quasi-unique responses": it should be impossible for an adversary to create two valid transcripts that differ only in the responses. Applying the Fiat-Shamir transform to a relaxed Σ-protocol with quasi-unique responses results in a relaxed NIZK proof, i.e., a non-interactive protocol that satisfies classical completeness, unbounded non-interactive zero-knowledge and the following relaxed definition of simulation soundness:

Definition 3 (Relaxed unbounded simulation soundness.). *There exists a PPT simulator* S *such that for all PPT adversaries* A*,*

$$\Pr\left[\mathcal{V}^{\mathsf{S}_1}(x^*,\pi^*) = 1 \wedge x^* \notin \bar{L} \wedge (x^*,\pi^*) \notin Q \ : \ (x^*,\pi^*) \leftarrow \mathsf{A}^{\mathsf{S}_1,\mathsf{S}_2'}(1^\lambda)\right]$$

is negligible, where Q *is the set of tuples* (x,π) *where* A *made a query* $\mathsf{S}_2(x)$ *and obtained response* π*.*

It is also possible to obtain relaxed unbounded simulation soundness using an OTS scheme with the Fiat-Shamir transform. A formal description and full proof of the construction can be found in the work by Boschini et al. [11].

To instantiate such protocols over lattices, consider the languages $(L,\bar{L})$ associated with the following relations:

$$R = \left\{((\mathbf{A},\mathbf{U}),(\mathbf{S},\mathbf{1})) \in \mathcal{R}_q^{\ell\times m} \times \mathcal{R}_q^{1\times\ell} \times \mathcal{R}_q^{m} \times \{\mathbf{1}\} : \mathbf{A}\mathbf{S} = \mathbf{U}, \|\mathbf{S}\| \le N\right\}$$

$$\bar{R} = \left\{((\mathbf{A},\mathbf{U}),(\bar{\mathbf{S}},\bar{\mathbf{c}})) \in \mathcal{R}_q^{\ell\times m} \times \mathcal{R}_q^{1\times\ell} \times \mathcal{R}_q^{m} \times \bar{\mathcal{C}} : \mathbf{A}\bar{\mathbf{S}} = \bar{\mathbf{c}}\mathbf{U}, \|\mathbf{S}\| \le \bar{N}_2,\ \|\mathbf{S}\|_\infty \le \bar{N}_\infty\right\}$$

where $0 < N \le \bar{N}_2$, $0 < \bar{N}_\infty$ and, if the set of the challenges used in the protocol is $\mathcal{C}$, the set of relaxed challenges is $\bar{\mathcal{C}} = \{\mathbf{c}-\mathbf{c}' \ : \ \mathbf{c},\mathbf{c}' \in \mathcal{C}\}$. Finding a witness $(\mathbf{S},\mathbf{c})$ for an element $(\mathbf{A},\mathbf{U})$ of the language $\bar{L}$ is hard under the computational assumption that Ring-SIS$_{\bar{N}}$ is hard. In the relaxed Σ-protocol for $L,\bar{L}$, the prover $\mathcal{P}$ samples a masking vector $\mathbf{Y} \xleftarrow{\$} \mathcal{D}_\sigma^m$ and sends $\mathbf{T} = \mathbf{A}\mathbf{Y}$ to the verifier $\mathcal{V}$. Next, $\mathcal{V}$ samples a challenge $\mathbf{c} \in \mathcal{C}$ and sends it back to $\mathcal{P}$. The prover constructs $\mathbf{Z} = \mathbf{Y} + \mathbf{c}\mathbf{S}$ and, depending on rejection sampling (see [30, Theorem 4.6]), either aborts or sends it to $\mathcal{V}$. The verifier accepts if $\mathbf{A}\mathbf{Z}-\mathbf{c}\mathbf{U} = \mathbf{T}$ and $\|\mathbf{Z}\| \le 1.05\sigma\sqrt{mn} =: N_2$, $\|\mathbf{Z}\|_\infty \le 8\sigma =: N_\infty$. The zero-knowledge property is guaranteed by rejection sampling. A standard deviation $\sigma = 12T$, where T is a bound on the norm of $\mathbf{c}\mathbf{S}$ obtained from N, guarantees that the prover outputs something with probability greater than $(1-2^{100})/e$ (cf. [30, Theorem 4.6]). Setting $\bar{N}_2 = 2N_2 = 2.1\sigma\sqrt{mn}$ and $\bar{N}_\infty = 2N_\infty = 16\sigma$ allows to prove that this is a relaxed Σ-protocol.

The proof-system we introduced can be adapted to prove that a component $\mathbf{s}_i$ of $\mathbf{S}$ is in a subring $\mathcal{R}_q^{(2^{K_m})}$ by using as challenge space $\mathcal{C} = \mathcal{R}_3^{(2^{K_c})}$, that is a subset of $\mathcal{R}_q^{(2^{K_m})}$ when $K_m \ge K_c$ and sampling the i-th element of the "masking" vector $\mathbf{Y}$ from $\mathcal{R}_q^{(2^{K_m})}$. Hence the output vector $\mathbf{Z} = \mathbf{S}\mathbf{c} + \mathbf{Y}$ is such that $\mathbf{z}_i \in \mathcal{R}_q^{(2^{K_m})}$. The verifier has to check also this latter condition before accepting.

2.8 Relaxed Signatures

Boschini et at. [11] introduced a new lattice-based relaxed signature scheme, i.e., a signature ($\mathsf{SParGen}, \mathsf{SKeyGen}, \mathsf{Sign}, \mathsf{SVerify}$) where the verification algorithm is relaxed to accept signature on messages coming from a set $\bar{\mathcal{M}}$ larger than the set $\mathcal{M}$ of signed messages. The signature is proved unforgeable under a relaxed notion of unforgeability under chosen-message attacks that includes as a forgery a signature on a message in $\bar{\mathcal{M}}$ that is the image of a message in $\mathcal{M}$ through some function f that was not signed by the signing oracle. The relaxation in the definition is necessary in order to combine the signature with the relaxed Σ-protocol (see Section 2.7).

Given that we reduce the unforgeability of the group signature directly to the hardness of Assumption 1, we do not discuss security of the signature here. We only remark that we use a different set of messages, namely $\mathcal{U} = \mathcal{R}_3^{(16)}$, while the original lattice instantiation signs messages composed by a small polynomial and a bit-string. When using it in the group signature, the small polynomial $\mathbf{m} \in \mathcal{U}$ encodes a user's identity, but there is no need for the bit string. Therefore, we substitute the output of the hash of the bit-string with a constant polynomial $\mathbf{u}$ chosen uniformly at random in $\mathcal{R}_q$ during the key generation and sign only messages in $\mathcal{M} = \mathcal{U}$. The modified scheme is trivially still unforgeable under Assumption 3 in the Random Oracle Model.

Parameters Generation. The parameters *spar* are generated by $\mathsf{SParGen}(1^\lambda)$ and include $(n, q, m, \sigma_t, \sigma, r, N, N', C', \mathbf{C})$ where: n is a power of 2, q is a prime, $q \equiv 5 \bmod 8$, m determines the gadget vector $\mathbf{G}$ in Theorem 1, σ_t is standard deviation of the distribution of the trapdoor, $\sigma = q^{1/m} \frac{\sigma_t}{\sqrt{\pi}} \sqrt{n} \cdot (\sqrt{2} + \sqrt{m} + \log(n))$ is the standard deviation of the Gaussian from which signatures are sampled, r bounds the norm of the polynomial part of the messages in $\bar{\mathcal{U}} = \mathcal{R}_r^{(16)}$, $N = 1.05\sigma\sqrt{n(2m+2)}$ bounds the norm of a signature output by Sign, $N' > N$ and $C' \geq 1$ define the set of valid signatures $\bar{\Sigma}$, and $\mathbf{C}$ is uniformly random matrix in $\mathcal{R}_q^{1\times m}$.

Key generation. The signer selects a uniformly random matrix $\mathbf{A} = \begin{bmatrix}\mathbf{a}\ \mathbf{1}\end{bmatrix}$ in $\mathcal{R}_q^{1\times 2}$ and an element $\mathbf{u} \xleftarrow{\$} \mathcal{R}_q$ as verification key and a matrix with small coefficients $\mathbf{X} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q, \sigma_t}^{2\times m}$ as secret signing key. The public verification key is the vector $\mathbf{V} = \begin{bmatrix}\mathbf{A}\ \mathbf{B}\ \mathbf{C}\end{bmatrix} = \begin{bmatrix}\mathbf{A}\ \mathbf{AX} + \mathbf{G}\ \mathbf{C}\end{bmatrix} \in \mathcal{R}_q^{1\times(2+2m)}$.

Signing. If $M = \mathbf{m} \notin \mathcal{M}$ abort. Otherwise, the signer computes $\mathbf{S} \leftarrow \mathsf{SampleD}$ $(\begin{bmatrix}\mathbf{A}\ \mathbf{B}\ \mathbf{C} + \mathbf{mG}\end{bmatrix}, \mathbf{u}, \sigma)$ (see Lemma 4) and outputs a signature $sig = (\mathbf{1}, \begin{bmatrix}\mathbf{S}\ ;\ \mathbf{0}\end{bmatrix}, \mathbf{1})$. The entry $(\mathbf{m}, sig)$ is stored so that if a signature on $\mathbf{m}$ is queried twice, the algorithm outputs always *sig*.

Verification. Verification of a signature $sig = (\mathbf{c}_1, \mathbf{S}, \mathbf{c}_2)$ on message $M = \mathbf{m}$ returns 1 if $\begin{bmatrix}\mathbf{A}\ \mathbf{B}\ \mathbf{c}_1\mathbf{C} + \mathbf{mG}\ \mathbf{1}\end{bmatrix}\mathbf{S} = \mathbf{c}_2\mathbf{u}$, if the message $M \in \bar{\mathcal{M}}$, and if the signature $sig \in \bar{\Sigma} = \{(\mathbf{c}_1, \mathbf{S}, \mathbf{c}_2) \in \bar{\mathcal{C}} \times \mathcal{R}_q^{3+2m} \times \mathcal{R}_q :\ \|\mathbf{S}\| \leq N' \ \wedge\ \|\mathbf{c}_2\| \leq C'\}$. Otherwise, it returns 0.

The relaxed signature scheme is f-uf-cma secure w.r.t. the message relaxation function $f(\mathbf{m}) = \{(\mathbf{mc})\ :\ \mathbf{c} \in \bar{\mathcal{C}}\}$.

Theorem 3. *An algorithm* A *that breaks the f-uf-cma unforgeability of the relaxed signature scheme in time t and probability ϵ_A can break the Assumption 3 in time t with probability ϵ_A in the Random Oracle Model.*

To prove knowledge of a signature on a message $\mathbf{m}$ without revealing $\mathbf{m}$, Boschini et al. combine the relaxed signature, a relaxed commitment and the relaxed Σ-protocol, where the commitment is used to hide the part of the verification key of the signature that depends on $\mathbf{m}$. Let $\mathbf{S} = \begin{bmatrix}\mathbf{S}_1\ ;\ \mathbf{S}_2\ ;\ \mathbf{S}_3\ ;\ \mathbf{1}\end{bmatrix}$ be a signature on $\mathbf{m}$ w.r.t. the public key $spk = \begin{bmatrix}\mathbf{A}\ \mathbf{B}\ \mathbf{C}\end{bmatrix}$. To hide the part of the verification equation of the signature that depends on $\mathbf{m}$, Boschini et al present the following trick. First, construct $\mathbf{F} = \mathbf{b}^{-1}(\mathbf{C} + \mathbf{mG} + \mathbf{E})$ choosing random $\mathbf{E} \xleftarrow{\$} \mathcal{R}_3^{1\times m}$ and $\mathbf{b} \xleftarrow{\$} \mathcal{R}_3$. Assuming Assumption 1 is hard or using complexity leveraging and assuming the hardness of Ring-LWE (cf. Section 2.4), we have that $\mathbf{F}$ hides $\mathbf{m}$[3] Then, set $\mathbf{S}_\mathrm{s}$ to be $\mathbf{S}_\mathrm{s} = \left[\mathbf{S}_1 \ ; \mathbf{S}_2 \ ; \mathbf{b}\mathbf{S}_3 \ ; -\mathbf{E}\mathbf{S}_3\right]$. It is easy to see that $\mathbf{s}_\mathrm{s}$ satisfies $\left[\mathbf{A}\ \mathbf{B}\ \mathbf{F}\ \mathbf{1}\right] \mathbf{S}_\mathrm{s} = \mathbf{u}$.

3 Relaxed Partial Verifiable Encryption

Lyubashevsky and Neven [32] defined a relaxed verifiable encryption as a scheme to encrypt a witness w of $x \in L$ such that decryption of a valid ciphertext is guaranteed to yield a witness $\bar{w}$ in the relaxed language such that $(x, \bar{w}) \in \bar{R}$.

The straightforward combination with the relaxed signature and commitment scheme of Boschini et al. [11] does not yield a particularly efficient group signature scheme, however, because the Lyubashevsky-Neven verifiable encryption scheme encrypts and recovers the *full* witness. A group signature typically consists of a verifiable encryption of the user's identity together with a proof that the user knows a valid signature on the encrypted identity by the group manager. The verifiable encryption as defined by Lyubashevsky and Neven would therefore encrypt both the user's identity and the signature on it, which unnecessarily blows up the size of the verifiable ciphertext. Even when using a commitment to the user's identity to separate the proof of knowledge of the signature from the verifiable encryption, the ciphertext will encrypt the user's identity as well as the opening information to the commitment.

We therefore introduce a variant of the Lyubashevsky-Neven relaxed verifiable encryption scheme called relaxed *partial* verifiable encryption that, rather than decrypting the full witness $\bar{w}$, recovers only a function of that witness $g(\bar{w})$ while proving knowledge of the full witness $\bar{w}$. When constructing a group signature case, we will use a function g that outputs just the user's identity.

3.1 Definition of Relaxed Partial Verifiable Encryption

Our general definition of relaxed partial verifiable encryption are inspired by the definition of relaxed verifiable encryption by Lyubashevsky and Neven [32] and of verifiable encryption by Camenisch and Shoup [14]. Let L be a language with witness relation R and let $\bar{L} \supseteq L$ be a relaxed language with relaxed relation $\bar{R} \supseteq R$. Let $\bar{R} \subseteq \bar{L} \times \bar{W}$ and let $g : \bar{W} \to D$ be a function.

Given relations R, $\bar{R}$ and function g, a *relaxed partial verifiable encryption scheme* is composed by four algorithms $(\mathsf{EKeyGen}, \mathsf{Enc}, \mathsf{EVerify}, \mathsf{Dec})$. The key generation algorithm $\mathsf{EKeyGen}(1^\lambda)$ outputs a pair of keys (epk, esk). The encryption algorithm $\mathsf{Enc}(epk, x, w, \ell)$, where $(x, w) \in R$ and $\ell \in \{0,1\}^*$ is an encryption label, returns a ciphertext t and a proof $\pi = (\alpha, \beta, \gamma)$. Verification $\mathsf{EVerify}(epk, x, t, \pi, \ell)$ returns 1 if π shows that t is a valid ciphertext w.r.t. x and epk with label ℓ, and returns 0 otherwise. Finally, the decryption algorithm $\mathsf{Dec}(esk, x, t, \pi, \ell)$ returns a value M or a failure symbol $\perp$.

[3] Boschini et al. proved that, for $\mathcal{U} \subset \mathcal{R}_3^{(16)}$, this is actually a relaxed commitment scheme. We do not need the relaxed binding property, hence we can choose a larger set of messages (as long as it still guarantees the hiding property).

Correctness The scheme is correct if $\mathsf{Pr}\left[\mathsf{Dec}(esk, x, \mathsf{Enc}(epk, x, w, \ell)) = g(w)\right] = 1$ for all keys $(epk, esk) \leftarrow \mathsf{EKeyGen}(1^\lambda)$, all $(x, w) \in R$, and all $\ell \in \{0,1\}^*$.

Completeness The scheme satisfies completeness if $\mathsf{Pr}[\mathsf{EVerify}(epk, \mathsf{Enc}(epk, x, w, \ell), \ell) = 1] = 1$ for all keys $(epk, esk) \leftarrow \mathsf{EKeyGen}(1^\lambda)$, all $(x, w) \in R$, and all $\ell \in \{0,1\}^*$.

Special soundness Special soundness implies that a valid proof π is a proof of knowledge of a valid witness $\bar{w}$ for the relation $\bar{R}$ and that decryption of the ciphertext t returns $g(\bar{w})$. More specifically, for all PPT adversaries A there exists a PPT extractor E such that the following probability is negligible:

$$\mathsf{Pr}\left[\begin{array}{c} b = b' = 1 \;\wedge\; \beta \neq \beta' \;\wedge \\ \big(\; \mathsf{Dec}(esk, x, t, \ell) \neq g(\bar{w}) \\ \vee\; (x, \bar{w}) \notin \bar{R} \;\big) \end{array} \;:\; \begin{array}{c} (epk, esk) \leftarrow \mathsf{EKeyGen}(1^\lambda), \\ (x, t, (\alpha, \beta, \gamma, \beta', \gamma'), \ell) \leftarrow \mathsf{A}(epk, esk), \\ b \leftarrow \mathsf{EVerify}(epk, x, t, (\alpha, \beta, \gamma), \ell), \\ b' \leftarrow \mathsf{EVerify}(epk, x, t, (\alpha, \beta', \gamma'), \ell)), \\ \bar{w} \leftarrow \mathsf{E}(epk, esk, x, t, (\alpha, \beta, \gamma, \beta', \gamma'), \ell) \end{array}\right].$$

Chosen-ciphertext simulatability There exists a simulator S that outputs ciphertexts indistinguishable from honestly generated ones, i.e., the following probability is negligible:

$$\left|\, \mathsf{Pr}\left[b = b' \;:\; \begin{array}{l} b \xleftarrow{\$} \{0,1\},\; (epk, esk) \leftarrow \mathsf{EKeyGen}(1^\lambda), \\ (st, x, w, \ell) \leftarrow \mathsf{A}^{\mathsf{Dec}(esk,\cdot,\cdot,\cdot,\cdot)}(epk), \\ (t_0, \pi_0) \leftarrow \mathsf{Enc}(epk, x, w),\; (t_1, \pi_1) \leftarrow \mathsf{S}(epk, x, \ell), \\ b' \leftarrow \mathsf{A}^{\mathsf{Dec}(esk,\cdot,\cdot,\cdot,\cdot)}(st, t_b, \pi_b) \end{array} \right] - \tfrac{1}{2} \,\right|,$$

where A cannot query its Dec oracle on (x, t_b, π_b, ℓ).

Observe that our definition of Special Soundness hardwires the use of Fiat-Shamir in the general construction. It is possible to give a more general definition of Special Soundness adapting the definition of weak simulation extractability By Faust et al. [19], but such a definition would be beyond the scope of this paper.

3.2 Relaxed Partial Verifiable Encryption over Lattices

Let L and $\bar{L}$ be a language and its relaxed version defined w.r.t. the following relations

$$R_{\text{ve}} = \left\{ \begin{array}{c} ((\mathbf{A}, \mathbf{U}), (\mathbf{m}, \mathbf{S}, \mathbf{1})) \in \\ (\mathcal{R}_q^{\ell_1 \times (\ell_2+1)} \times \mathcal{R}_q^{\ell_1}) \times (\mathcal{U} \times \mathcal{R}_q^{\ell_2} \times \{\mathbf{1}\}) \end{array} \;:\; \mathbf{A} \begin{bmatrix} \mathbf{m} \\ \mathbf{S} \end{bmatrix} = \mathbf{U} \bmod q \wedge \; \|\mathbf{S}\| \leq N \right\}$$

$$\bar{R}_{\text{ve}} = \left\{ \begin{array}{c} ((\mathbf{A}, \mathbf{U}), (\bar{\mathbf{m}}, \bar{\mathbf{S}}, \bar{\mathbf{c}})) \in \\ (\mathcal{R}_q^{\ell_1 \times (\ell_2+1)} \times \mathcal{R}_q^{\ell_1}) \times (\bar{\mathcal{U}} \times \mathcal{R}_q^{\ell_2} \times \bar{\mathcal{C}}) \end{array} \;:\; \mathbf{A} \begin{bmatrix} \bar{\mathbf{m}} \\ \bar{\mathbf{S}} \end{bmatrix} = \bar{\mathbf{c}}\mathbf{U} \bmod q \wedge \; \|\bar{\mathbf{S}}\| \leq \bar{N} \right\}$$

for some sets $\mathcal{U}, \bar{\mathcal{U}}, \bar{\mathcal{C}} \subseteq \mathcal{R}_q$ and some integers $\ell_1, \ell_2, N, \bar{N} > 0$.

We will construct a relaxed partial verifiable encryption scheme for relations R_{ve} and $\bar{R}_{\text{ve}}$ and function $g((\bar{\mathbf{m}}, \bar{\mathbf{S}}, \bar{\mathbf{c}})) = \bar{\mathbf{m}}/\bar{\mathbf{c}} \bmod q$. Our scheme is a modified version of the "multi-shot" chosen-ciphertext secure verifiable encryption scheme of Lyubashevsky-Neven. The multi-shot scheme involves multiple parallel repetitions of the proof with sub-exponential challenge set sizes, and decryption takes strictly sub exponential time (as opposed to expected polynomial time for the one-shot scheme).

Rather than producing one big proof of knowledge of the terms in relation R_{ve}, we split it into two proofs, one for each term. The first proof only contains the ciphertext equations and is repeated multiple times with a sub-exponential challenge set to enable efficient decryption. The second includes the relation equation as well as the ciphertext, proving that the encrypted plaintext is derived from a valid witness. The latter proof uses an exponential-size challenge set, so that it doesn't need to be repeated. Let p and q be two public primes with $p > 2$.

Key Generation. The recipient generates two key pairs for Ring-LWE encryption [33], but discards the secret key of the second pair. It samples $\mathbf{s}_1, \mathbf{d}_1, \mathbf{s}_2, \mathbf{d}_2 \xleftarrow{\$} \mathcal{R}_3$ and $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$, and computes $\mathbf{t}_1 = \mathbf{a}\mathbf{s}_1 + \mathbf{d}_1 \bmod q$ and $\mathbf{t}_2 = \mathbf{a}\mathbf{s}_2 + \mathbf{d}_2 \bmod q$. The public key is $epk = (p, q, \mathbf{a}, \mathbf{t}_1, \mathbf{t}_2)$, the secret key is $esk = \mathbf{s}_1$.

Encryption. Given a witness $(\mathbf{m}, \mathbf{S}, \mathbf{1})$ for language member $(\mathbf{A}, \mathbf{U})$ in the relation R_{ve}, the algorithm Enc uses the Naor-Yung technique [36] by encrypting $\mathbf{m}$ twice using standard Ring-LWE encryption under public keys $\mathbf{t}_1$ and $\mathbf{t}_2$. More precisely, it samples $\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{f}_1, \mathbf{f}_2 \xleftarrow{\$} \mathcal{R}_3$ and sets $\mathbf{v}_1 = p(\mathbf{a}\mathbf{r} + \mathbf{e}_1) \bmod q$, $\mathbf{w}_1 = p(\mathbf{t}_1\mathbf{r} + \mathbf{f}_1) + \mathbf{m} \bmod q$, $\mathbf{v}_2 = p(\mathbf{a}\mathbf{r} + \mathbf{e}_2) \bmod q$, and $\mathbf{w}_2 = p(\mathbf{t}_2\mathbf{r} + \mathbf{f}_2) + \mathbf{m} \bmod q$.

Then, letting $\mathbf{A}_1$ be the first column of the matrix $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in relation R_{ve}, it constructs a NIZK proof Π_1 using the scheme from Section 2.7 for the relation

$$\begin{bmatrix} \mathbf{0} & p\mathbf{a} & p & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}^{1\times\ell_2} \\ \mathbf{1} & p\mathbf{t}_1 & \mathbf{0} & p & \mathbf{0} & \mathbf{0} & \mathbf{0}^{1\times\ell_2} \\ \mathbf{0} & p\mathbf{a} & \mathbf{0} & \mathbf{0} & p & \mathbf{0} & \mathbf{0}^{1\times\ell_2} \\ \mathbf{1} & p\mathbf{t}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & p & \mathbf{0}^{1\times\ell_2} \\ \mathbf{A}_1 & \mathbf{0}^{\ell_1\times 1} & \mathbf{0}^{\ell_1\times 1} & \mathbf{0}^{\ell_1\times 1} & \mathbf{0}^{\ell_1\times 1} & \mathbf{0}^{\ell_1\times 1} & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \mathbf{r} \\ \mathbf{e}_1 \\ \mathbf{f}_1 \\ \mathbf{e}_2 \\ \mathbf{f}_2 \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{w}_1 \\ \mathbf{v}_2 \\ \mathbf{w}_2 \\ \mathbf{U} \end{bmatrix}, \tag{1}$$

whereby it uses the challenge set $\mathcal{C}_1 = \{\mathbf{c} \in \mathcal{R}_3 \mid \|\mathbf{c}\|_1 \leq 32\}$.

To enable Lyubashevsky-Neven's multi-shot decryption technique without having to repeat the above proof multiple times, the encryptor again uses the relaxed NIZK proof of Section 2.7 to construct a separate proof Π_2 for the relation

$$\begin{bmatrix} \mathbf{0} & p\mathbf{a} & p & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & p\mathbf{t}_1 & \mathbf{0} & p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & p\mathbf{a} & \mathbf{1} & \mathbf{0} & p & \mathbf{0} \\ \mathbf{1} & p\mathbf{t}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & p \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \mathbf{r} \\ \mathbf{e}_1 \\ \mathbf{f}_1 \\ \mathbf{e}_2 \\ \mathbf{f}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{w}_1 \\ \mathbf{v}_2 \\ \mathbf{w}_2 \end{bmatrix}, \tag{2}$$

whereby it includes $epk, (\mathbf{A}, \mathbf{U}), (\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2), \Pi_1, \ell$ in the Fiat-Shamir hash. To obtain efficient decryption but keep the soundness error negligible, this proof is repeated $l = 11$ times with challenge set $\mathcal{C}_2 = \mathcal{R}_3^{(16)}$. The algorithm outputs ciphertext $(\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2)$ and proof (Π_1, Π_2).

Verification. The verification algorithm $\mathsf{EVerify}((p, q, \mathbf{a}, \mathbf{t}_1, \mathbf{t}_2), (\mathbf{A}, \mathbf{U}), (\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2, \Pi_1, \Pi_2), \ell)$ checks that Π_1 and Π_2 are valid relaxed NIZK proofs for the relations of Equations (1) and (2), including the correct arguments $epk, (\mathbf{A}, \mathbf{U}), (\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2), \Pi_1, \ell$ in the Fiat-Shamir hash of Π_2.

Decryption. The decryption algorithm $\mathsf{Dec}(\mathbf{s}_1, (\mathbf{A}, \mathbf{U}), (\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2), (\Pi_1, \Pi_2), \ell)$ first checks that the proofs are valid using the verification algorithm above, returning $\perp$ if it is not valid. It then decrypts the cihpertext by applying the

Lyubashevsky-Neven multi-shot decryption on proof $\Pi_2 = (\mathbf{Y}^{(1)}, \mathbf{c}^{(1)}, \mathbf{Z}^{(1)}, \ldots, \mathbf{Y}^{(l)}, \mathbf{c}^{(l)}, \mathbf{Z}^{(l)})$ by, for $i = 1, \ldots, l$, going over all challenges $\mathbf{c}' \in \mathcal{C}_2$ to try to decrypt $(\bar{\mathbf{c}}\mathbf{v}, \bar{\mathbf{c}}\mathbf{w}_1)$ as a Ring-LWE ciphertext, where $\bar{\mathbf{c}} = \mathbf{c}^{(i)} - \mathbf{c}'$. It does so by computing $\bar{\mathbf{m}}' = (\mathbf{w}_1 - \mathbf{v}_1\mathbf{s}_1)\bar{\mathbf{c}} \bmod q$, checking that $\|\bar{\mathbf{m}}'\|_\infty < q/2C$ where C is as defined in Lemma 5, and if so, compute $\bar{\mathbf{m}} = \bar{\mathbf{m}}' \bmod p$ and return $\bar{\mathbf{m}}/\bar{\mathbf{c}} \bmod q$; otherwise, it returns $\perp$.

Decryption Runtime. Decryption terminates in time at most 2^{26}. Indeed, if the ciphertext is honestly generated the algorithm needs to guess the challenge only once. On the other hand, for a dishonestly generated ciphertext the probability that verification succeeds and still decryption fails is negligible. Indeed, if the adversary could answer only one challenge $\mathbf{c}$, when making the random oracle queries the probability of hitting always $\mathbf{c}$ would be $1/(\ell \cdot |\mathcal{C}_2|)$. Hence, a second challenge exists w.h.p. and decryption requires to guess a challenge $\mathbf{c}'$ at most $|\mathcal{C}_2| \leq 2^{26}$ times.

Remark that the decryption does not recover the full witness: the algorithm decrypts the ciphertext, but it does not recover the randomness used to generate it or the vector $\mathbf{S}$. Moreover, differently from Lyubashevsky-Neven construction, in our case the relation $\mathbf{A}\begin{bmatrix}\mathbf{m}\\ \mathbf{S}\end{bmatrix} = \mathbf{U}$ holds modulo q, while in the original scheme it has to hold modulo p. We show the correctness of the scheme using Lemma 5, which is a variant of a result by Lyubashevsky and Neven [32, Lemma 3.1]. In this lemma we show that, for some choice of the parameters, the decryption always return the same value $\bar{\mathbf{m}}/\bar{\mathbf{c}}$ over the ring $\mathcal{R}_q$. This is slightly different from the original decryption algorithm, as in the original scheme it was enough for decryption to return the same $\bar{\mathbf{m}}/\bar{\mathbf{c}}$ modulo p.

Lemma 5. *Let* $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$, *and* $\mathbf{t} = \mathbf{a}\mathbf{s} + \mathbf{d}$ *where* $\mathbf{s}, \mathbf{d} \xleftarrow{\$} \mathcal{R}_3$. *If there exist* $\bar{\mathbf{r}}$, $\bar{\mathbf{e}}$, $\bar{\mathbf{f}}$, $\bar{\mathbf{m}}$, $\bar{\mathbf{c}}$ *such that*

$$p(\mathbf{a}\bar{\mathbf{r}} + \bar{\mathbf{e}}) = \bar{\mathbf{c}}\mathbf{v} \ \mathit{mod}\ q \quad \mathit{and} \quad p(\mathbf{t}\bar{\mathbf{r}} + \bar{\mathbf{f}}) + \bar{\mathbf{m}} = \bar{\mathbf{c}}\mathbf{w} \ \mathit{mod}\ q \tag{3}$$

and $\|p(\bar{\mathbf{r}}\mathbf{d} + \bar{\mathbf{f}} - \bar{\mathbf{e}}\mathbf{s}) + \bar{\mathbf{m}}\|_\infty < q/2C$ *and* $\|\bar{\mathbf{m}}\|_\infty < p/2C$, *where* $C = \max_{\bar{\mathbf{c}} \in \bar{\mathcal{C}}} \|\bar{\mathbf{c}}\|_1$ $= \max_{\bar{\mathbf{c}}, \bar{\mathbf{c}}' \in \mathcal{C}} \|\bar{\mathbf{c}} - \bar{\mathbf{c}}'\|_1$, *then*

1. $\|(\mathbf{w} - \mathbf{v}\mathbf{s})\mathbf{c}' \ \mathit{mod}\ q\|_\infty < q/2C$ *and* $\|(\mathbf{w} - \mathbf{v}\mathbf{s})\mathbf{c}' \ \mathit{mod}\ q \ \mathit{mod}\ p\|_\infty < p/2C$
2. *for any* $\bar{\mathbf{c}}' \in \bar{\mathcal{C}}$ *such that* $\|(\mathbf{w} - \mathbf{v}\mathbf{s})\mathbf{c}' \ \mathit{mod}\ q\|_\infty < q/2C$ *and* $\|(\mathbf{w} - \mathbf{v}\mathbf{s})\mathbf{c}' \ \mathit{mod}\ q \ \mathit{mod}\ p\|_\infty < p/2C$ *we have* $(\mathbf{w} - \mathbf{v}\mathbf{s})\bar{\mathbf{c}}' \ \mathit{mod}\ q \ \mathit{mod}\ p/\bar{\mathbf{c}}' = \bar{\mathbf{m}}/\bar{\mathbf{c}}$.

Proof. The proof is a simple verification of the claims and it is very similar to the proof of Lemma 3.1 in [32], hence we omit it.

Hence, for decryption to be correct, we must choose parameters that guarantee that the values decrypted from Π_2 using $\mathbf{s}_i$ for $i = 1, 2$ satisfy $\|p(\bar{\mathbf{r}}_i\mathbf{d}_i + \bar{\mathbf{f}}_i - \bar{\mathbf{e}}_i\mathbf{s}_i) + \bar{\mathbf{m}}\|_\infty < q/2C$ and $\|\bar{\mathbf{m}}_i\|_\infty < p/2C$, i.e., p, q and n should be such that $16\sigma_2(2np + p + 1) < q/2C$ and $16\sigma_2 < p/2C$, where $C \leq 64$ as challenges come from $\mathcal{R}_3^{(16)}$. We enforce this condition on both ciphertexts to guarantee decryption to work using either $\mathbf{s}_1$ or $\mathbf{s}_2$. This allows to prove CCA simulatability following the Naor-Young paradigm [36].

In the next lemma, we prove that with high probability the $\bar{\mathbf{m}}/\bar{\mathbf{c}}$ returned by decryption is equal to the polynomial $\bar{\mathbf{m}}'/\bar{\mathbf{c}}'$ returned from an extractor for Π_2.

The proof of this lemma consists only of a plain computation of the probability, and can be found in the full version of the paper.

Lemma 6. *Let $\bar{\mathbf{m}}$ and $\bar{\mathbf{c}}$ be the output of the decryption and $\bar{\mathbf{m}}', \bar{\mathbf{c}}'$ be the values extracted from Π_1. Then with probability at least $1 - 2^{-35928}$, over the choice of the opening key $\mathbf{t}$, $\bar{\mathbf{m}}/\bar{\mathbf{c}} = \bar{\mathbf{m}}'/\bar{\mathbf{c}}'$ (where parameters are set as in Table 1).*

Finally, for the CCA simulatability the proofs that we use in the scheme need to be unbounded simulation soundness. Following the same reasoning used in Lyubashevsky and Neven, we prove that Π_2 has quasi-unique responses, hence simulation soundness. Indeed, breaking quasi-uniqueness means finding $\mathbf{z} \neq \mathbf{z}'$ with ℓ_∞ norm less than $8\sigma_2$ such that $\mathbf{Mz} = \mathbf{Mz}' \bmod q$, where with $\mathbf{M}$ we mean the matrix in 2. Thus, either there is a non-zero tuple $(\mathbf{y}_1, \mathbf{y}_2) \in \mathcal{R}_q$ with ℓ_∞ norm less than $16\sigma_2$ such that $p(\mathbf{a}\mathbf{y}_1 + \mathbf{y}_2) = 0 \bmod q$ or $p\mathbf{y}_1 + \mathbf{y}_2 = 0 \bmod q$. Imposing $p > 16\sigma_2$ and $16\sigma_2 p + 16\sigma_2 < q$ implies that the second equality is not possible. Also, setting $(32\sigma_2)^2 < q$, we can use a standard probabilistic argument to show that for all $\mathbf{y}_1$, $\mathbf{y}_2$ of ℓ_∞ norm less than $16\sigma_2$,

$$\mathsf{Pr}_{\mathbf{a} \in \mathcal{R}_q}\left[\mathbf{a}\mathbf{y}_1 + p\mathbf{y}_2 = 0 \bmod q\right] = 2^{-\Omega(n)} .$$

Therefore for almost all $\mathbf{a}$, there will not be a short solution $(\mathbf{y}_1, \mathbf{y}_2)$ that satisfies $\mathbf{a}\mathbf{y}_1 + p\mathbf{y}_2 = 0$. Observe that the same argument works for Π_1. Hence imposing the same inequalities on σ_1 yields simulation soundness also for Π_1, thus for the protocol (Π_1, Π_2).

Theorem 4. *If Ring-LWE$_{U(\mathcal{R}_q)}$ is hard and the relaxed NIZK proof system is unbounded non-interactive zero-knowledge and unbounded simulation soundness, the scheme* $(\mathsf{EKeyGen}, \mathsf{Enc}, \mathsf{EVerify}, \mathsf{Dec})$ *is a relaxed partial verifiable encryption scheme w.r.t. the function g.*

4 Group Signature Scheme

The combination of Boschini et al.'s relaxed signature scheme [11] with our relaxed partial verifiable encryption scheme yields an efficient group signature with practical parameters (see Section 4.2). Although the building blocks are "relaxed" schemes, the resulting group signature enjoys non-relaxed traceability. Indeed, the correctness of the verifiable encryption guarantees that when opening a signature, the recovered identity is in the original set of group members id (and not in the relaxed one).

4.1 A Lattice-Based Group Signature

Let $\mathcal{U} = \mathcal{R}_3^{(16)}$ be the set of possible user identities.

Parameters Generation. On input the security parameter λ, the algorithm runs the parameter generator of the signature scheme $par \leftarrow \mathsf{SParGen}(1^\lambda)$ and chooses integer p, q, n where p and q are prime and $p < q$. It outputs $gpar := (par, p, q, n)$.

Group Manager Key Generation. The group manager generates the keys $gsk = \mathbf{X}$ and $gpk = (\begin{bmatrix}\mathbf{A}\ \mathbf{B}\ \mathbf{C}\ \mathbf{1}\end{bmatrix}, \mathbf{u})$ by running $\mathsf{SKeyGen}$ and choosing a random ring element $\mathbf{u} \xleftarrow{\$} \mathcal{R}_q$.

Opener Key Generation. The opener runs the key generation algorithm of the verifiable encryption scheme $\mathsf{EKeyGen}(1^\lambda)$ and returns the resulting key pair $(opk = epk, osk = esk)$.

User Key Generation. The group manager generates a signing key user identity $id = \mathbf{m} \in \mathcal{U} = \mathcal{R}_3^{(16)}$ by running $\mathsf{Sign}(gsk, \mathbf{m})$ to yield $(\mathbf{1}, \left[\mathbf{S}\ ;\ \mathbf{0}\right], \mathbf{1})$ as described in Section 2.8. Recall that $\mathbf{S}$ is a short vector so that $\left[\mathbf{A}\ \mathbf{B}\ \mathbf{C} + \mathbf{m}\mathbf{G}\right]\mathbf{S} = \mathbf{u} \bmod q$. It then returns $usk := \mathbf{S}$.

Signing Algorithm. The user first generates a key one-time signature key pair $(sk, vk) \leftarrow \mathsf{OTSGen}(1^\lambda)$. The user then blinds her identity $\mathbf{m}$ using the technique from Section 2.8 by choosing random $\mathbf{E} \xleftarrow{\$} \mathcal{R}_3^{1\times m}$ and $\mathbf{b} \xleftarrow{\$} \mathcal{R}_3$, and computing $\mathbf{F} = \mathbf{b}^{-1}(\mathbf{C}+\mathbf{m}\mathbf{G}+\mathbf{E})$. If $\mathbf{S} = \left[\mathbf{S}_1\ ;\ \mathbf{S}_2\ ;\ \mathbf{S}_3\right]$ with $\mathbf{S}_1 \in \mathcal{R}_q^{2\times 1}$ and $\mathbf{S}_2, \mathbf{S}_3 \in \mathcal{R}_q^{m\times 1}$, then we have that $\left[\mathbf{A}\ \mathbf{B}\ \mathbf{F}\ \mathbf{1}\right]\left[\mathbf{S}_1\ ;\ \mathbf{S}_2\ ;\ \mathbf{b}\mathbf{S}_3\ ;\ -\mathbf{E}\mathbf{S}_3\right] = \mathbf{u} \bmod q$. The user can therefore create a relaxed NIZK proof $\mathit{\Pi}_0$ for the relation

$$\begin{aligned} R_0 &= \left\{((\left[\mathbf{A}\ \mathbf{B}\ \mathbf{F}\ \mathbf{1}\right], \mathbf{u}), (\mathbf{T}_0, \mathbf{1}))\ :\ \left[\mathbf{A}\ \mathbf{B}\ \mathbf{F}\ \mathbf{1}\right]\mathbf{T}_0 = \mathbf{u}\ \wedge\ \|\mathbf{T}_0\| \le N_0\right\} \\ \bar{R}_0 &= \left\{((\left[\mathbf{A}\ \mathbf{B}\ \mathbf{F}\ \mathbf{1}\right], \mathbf{u}), (\bar{\mathbf{T}}_0, \bar{\mathbf{c}}))\ :\ \left[\mathbf{A}\ \mathbf{B}\ \mathbf{F}\ \mathbf{1}\right]\bar{\mathbf{T}}_0 = \bar{\mathbf{c}}\mathbf{u} \wedge\ \bar{\mathbf{c}} \in \bar{\mathcal{C}}_0\ \wedge\ \|\bar{\mathbf{T}}_0\| \le \bar{N}_0\ \right\} \end{aligned} \tag{4}$$

where she includes vk in the Fiat-Shamir hash. The parameters follow from rejection sampling (see Section 2.7): the noise vector is sampled from a Gaussian with standard deviation $\sigma_0 = 12T_0$, where T_0 is obtained from N_0 as a bound on the norm of $\mathbf{c}\mathbf{T}_0$ for $\mathbf{c} \in \mathcal{C}_0$, and $\bar{N}_0 = 2.1\sigma_0\sqrt{n(3+2m)}$. The challenge space is set to $\mathcal{C}_0 = \{\mathbf{c} \in \mathcal{R}_3\ :\ \|\mathbf{c}\|_1 \le 32\}$ so that the proof only needs to be repeated once, as indeed $|\mathcal{C}_0| > 2^{256}$.

Next, from the way $\mathbf{F}$ was computed, we have that $\left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right]\left[\mathbf{m}\ ;\ -\mathbf{b}\ ;\ \mathbf{E}^\top\right] = -\mathbf{C}^\top$. Setting $\mathbf{T}_{\text{ve}} = \left[-\mathbf{b}\ ;\ \mathbf{E}^\top\right]$ the prover can therefore use the verifiable encryption scheme to encrypt a witness of the languages with relations

$$R_{\text{ve}} = \left\{\begin{array}{c} ((\left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right], -\mathbf{C}^\top), (\mathbf{m}, \mathbf{T}_{\text{ve}}, \mathbf{1})) \in (\mathcal{R}_q^{m\times(m+2)} \times \mathcal{R}_q^m) \times (\mathcal{U} \times \mathcal{R}_q^{m+1} \times \{\mathbf{1}\}) \\ :\qquad \left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right]\begin{bmatrix}\mathbf{m} \\ \mathbf{T}_{\text{ve}}\end{bmatrix} = -\mathbf{C}^\top \bmod q\ \wedge\ \|\mathbf{T}_{\text{ve}}\| \le N_{\text{ve}} \end{array}\right\}$$

$$\bar{R}_{\text{ve}} = \left\{\begin{array}{c} ((\left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right], -\mathbf{C}^\top), (\bar{\mathbf{m}}, \bar{\mathbf{T}}_{\text{ve}}, \bar{\mathbf{c}})) \in (\mathcal{R}_q^{m\times(m+2)} \times \mathcal{R}_q^m) \times (\bar{\mathcal{U}} \times \mathcal{R}_q^{m+1} \times \bar{\mathcal{C}}_{\text{ve}}) \\ :\qquad \left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right]\begin{bmatrix}\bar{\mathbf{m}} \\ \bar{\mathbf{T}}_{\text{ve}}\end{bmatrix} = -\bar{\mathbf{c}}\mathbf{C}^\top \bmod q\ \wedge\ \|\bar{\mathbf{T}}_{\text{ve}}\| \le \bar{N}_{\text{ve}} \end{array}\right\}$$

The user runs the encryption algorithm $\mathsf{Enc}(opk, x, w, vk)$ with language member $x = (\left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right], -\mathbf{C}^\top)$, witness $w = (\mathbf{m}, \left[-\mathbf{b}\ ;\ \mathbf{E}^\top\right], \mathbf{1})$, and the verification key vk as the encryption label, to generate a ciphertext $t = (\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2)$ and proof $\pi = (\mathit{\Pi}_1, \mathit{\Pi}_2)$. The user then computes the one-time signature $ots \leftarrow \mathsf{OTSSign}(sk, (\mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{u}, \mathit{\Pi}_0, t, \pi, M))$ and returns the group signature $sig = (\mathbf{F}, \mathit{\Pi}_0, t, \pi, vk, ots)$.

Verification Algorithm. The verifier checks the one-time signature by running $\mathsf{OTSVf}(vk, (\mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{u}, \mathit{\Pi}_0, t, \pi, M), ots)$, checks the NIZK proof $\mathit{\Pi}_0$ in the group signature $sig = (\mathbf{F}, \mathit{\Pi}_0, t, \pi)$, making sure that vk is included in the Fiat-Shamir hash, and checks the encryption proof by running $\mathsf{EVerify}(opk, x, t, \pi, vk)$ with $x = (\left[\mathbf{G}^\top\ \mathbf{F}^\top\ \mathbb{I}_m\right], -\mathbf{C}^\top)$ and with vk as the encryption label. If all tests succeed then he outputs 1, else he outputs 0.

Opening Algorithm. The opener first verifies the group signature by running the $\mathsf{GVerify}$ algorithm above. If it is invalid, then the opener returns $\perp$, else it decrypts $\mathbf{m} \leftarrow \mathsf{Dec}(esk, x, t, \pi, vk)$ with x as above and returns $id = \mathbf{m}$.

To guarantee the correctness of the scheme, the norm bounds N_0, N_{ve} and $\bar{N}_{\mathrm{ve}}$ should be chosen carefully. First, as observed in Section 2.8, a honest $\mathbf{T}$ is generated as $\mathbf{T} = \begin{bmatrix}\mathbf{S}_1 \; \mathbf{S}_2 \; \mathbf{bS}_3 \; -\mathbf{ES}_3\end{bmatrix}$, where the vector $\mathbf{S} = \begin{bmatrix}\mathbf{S}_1 \; \mathbf{S}_2 \; \mathbf{S}_3\end{bmatrix} \in \mathcal{R}_q^{1\times(2+2m)}$ is sampled from a Gaussian with standard deviation σ. Hence it each of its components has norm bounded by $1.05\sigma\sqrt{n}$. Moreover, using the bounds in Lemma 2, it holds $\|\mathbf{bS}_3\| \le 8\sigma n\sqrt{m}$ and $\|-\mathbf{ES}_3\| \le \sqrt{\sum_{i=1}^{m}\|\mathbf{E}_i\mathbf{S}_{3,i}\|_2^2} \le 8\sigma n\sqrt{m}$. Hence we can set the bound N_0 to be:

$$N_0 = \sqrt{(2+m)(1.05\sigma\sqrt{n})^2 + m(8\sigma n)^2 + m(8\sigma n)^2}.$$

The value N_{ve} in R_{ve} bounds the norm of a vector of polynomials with coefficients in $\{0,1\}$ one of which is in $\mathcal{R}_3^{(16)}$, hence $N_{\mathrm{ve}} := \sqrt{256 + n(m^2+1)}$. Finally, the parameter $\bar{N}_{\mathrm{ve}}$ bounds the norm of what is returned extracting from the NIZK proof, hence it is computed from the standard deviation of the Gaussian distribution used in rejection sampling as explained in Section 2.7.

Theorem 5 (Traceability). *Our group signature scheme is traceable in the random-oracle model if Assumption 3 holds and the relaxed partial verifiable encryption scheme of Section 3 satisfies special soundness.*

Theorem 6 (CCA-Anonymity). *Our group signature scheme is CCA-anonymous in the random-oracle model if Assumption 1 holds, if the NIZK proof is statistical zero-knowledge and if the relaxed partial verifiable encryption scheme of Section 3 is chosen-ciphertext simulatable.*

As stated in Section 2.4, there are two ways to interpret Assumption 3 and Assumption 1, either as a quite strong interactive assumption, or as implied through a complexity leveraging argument by the Ring-LWE and the Ring-SIS assumptions, and by the Ring-LWE$_{m,D}$ assumption, respectively.

4.2 Practical Parameters and Storage Requirement

In Table 1 we give a set of practical parameters for different security requirements and all guaranteeing $\lambda = 80$ bits of security against quantum adversaries. Following the approach in Boschini et al. [11], we give the possibility to choose whether to base the security of the scheme on complexity leveraging or not. All parameters are computed w.r.t. fixed $n = 2^{11}$, $\sigma_t = 4$ and p a prime such that $\log p \le 2^{50}$. The second column contains the maximum value of the Hermite root factor computed for the Ring-SIS instance in Theorem 2. Given that not only Assumption 4, but also the hardness of finding a witness for an element of L in Section 2.7 is based on that, we decided to use it to have a hardness estimate even when relying only on the hardness of Assumption 3. The only difference with the other case (and the reason for which δ_s is different) is that when assuming complexity leveraging we need to compensate also for the tightness loss of the reductions in Section 2.4, while in the other case it is only necessary to compensate for the tightness loss in the proofs of Theorem 5 and 6. We recall

	Parameters						Sizes			
Compl. Lev.	δ_s	m	q	σ_0	σ_1	σ_2	gpk(MB)	usk(kB)	opk(kB)	sig(MB)
NO	1.00352	7	$\sim 2^{115}$	$2.891 \cdot 10^{17}$	$6.51 \cdot 10^4$	$2.13 \cdot 10^4$	0.501	122.95	88.32	0.91
YES	1.0014	22	$\sim 2^{116}$	$4.325 \cdot 10^{14}$	$9.36 \cdot 10^4$	$2.13 \cdot 10^4$	1.396	224.26	89.1	1.72

Table 1. Table of parameters for $n = 2^{11}$, $\sigma_t = 4$ and $p \sim 2^{50}$ for 2^{25} users.

that the most efficient scheme prior to ours [26] has signatures over 60 MB and public keys of 4.9 MB for a group size of only 2^{10} users for 80 bits of security. While they still have to deal with big lattices (dimensions: $n = 2^8$, $m = 2^{12}$), their coefficients are smaller than ours (bounded by $q = 2^8$), and this allow for more efficient computations.

Acknowledgements

The authors thank Vadim Lyubashevsky for many helpful discussions and the anonymous reviewers for the useful comments. This work was supported by the ERC under grant #321310 PERCY), the SNF under grant #200021_157080 (Efficient Lattice-Based Cryptographic Protocols) and the European Union's Horizon 2020 research and innovation program under grant #779391 (FutureTPM).

References

1. S. Agrawal, D. Boneh, X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO 2010*.
2. E. Alkim, L. Ducas, T. Pöppelmann P. Schwabe. Post-quantum Key Exchange - A New Hope, *USENIX Security Symposium 2016.*.
3. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*.
4. A. Bagherzandi, J. H. Cheon, S. Jarecki. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In *ACM CCS 2008*.
5. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
6. M. Bellare, D. Micciancio, B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. *EUROCRYPT 2003*.
7. M. Bellare, H. Shi, C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. *CT-RSA 2005*.
8. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *ASIACRYPT 2014*.
9. P. Bichsel, J. Camenisch, T. Groß, V. Shoup. Anonymous credentials on a standard Java card. In *ACM CCS 2009*.
10. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT 2005*.
11. C. Boschini, J. Camenisch, G. Neven. Relaxed lattice-based signatures with short zero-knowledge proofs. *Cryptology ePrint Archive, Report 2017/1123*, 2017.

12. E. F. Brickell, J. Camenisch, L. Chen. Direct anonymous attestation. In *ACM CCS 2004*.
13. J. Camenisch, G. Neven, M. Rückert. Fully anonymous attribute tokens from lattices. In *LNCS 2012*.
14. J. Camenisch, V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*.
15. M. Chase, A. Lysyanskaya. On signatures of knowledge. In *CRYPTO 2006*.
16. D. Chaum, E. van Heyst. Group signatures. In *EUROCRYPT 1991*.
17. L. Ducas, A. Durmus, T. Lepoint, V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In *CRYPTO 2013*.
18. L. Ducas, V. Lyubashevsky, T. Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT 2014*.
19. S. Faust, M. Kohlweiss, G. A. Marson, D. Venturi. On the non-malleability of the Fiat-Shamir transform. In *INDOCRYPT 2012*.
20. A. Fiat, A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *CRYPTO 1986*.
21. N. Gama, P. Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT 2008*.
22. C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *ACM STOC 2008*.
23. S. D. Gordon, J. Katz, V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT 2010*.
24. F. Laguillaumie, A. Langlois, B. Libert, D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT 2013*.
25. L. Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.
26. B. Libert, S. Ling, K. Nguyen, H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT 2016*.
27. B. Libert, F. Mouhartem, K. Nguyen. A lattice-based group signature scheme with message-dependent opening. In *ACNS 2016*.
28. S. Ling, K. Nguyen, H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC 2015*.
29. S. Ling, K. Nguyen, H. Wang, Y. Xu. Lattice-based group signatures: Achieving full dynamicity with ease. Cryptology ePrint Archive, Report 2017/353, 2017.
30. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*.
31. V. Lyubashevsky, D. Micciancio. Generalized compact Knapsacks are collision resistant. In *ICALP 2006*.
32. V. Lyubashevsky, G. Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT 2017*.
33. V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT 2010*.
34. V. Lyubashevsky, C. Peikert, O. Regev. A toolkit for ring-LWE cryptography. In *EUROCRYPT 2013*.
35. D. Micciancio, C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*.
36. M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *ACM STOC 1990*.
37. P. Q. Nguyen, J. Zhang, Z. Zhang. Simpler efficient group signatures from lattices. In *PKC 2015*.
38. M. Rückert. Lattice-based blind signatures. In *ASIACRYPT 2010*.

39. J. Stern. A new identification scheme based on syndrome decoding. In *CRYPTO 1993*.

40. K. Xagawa, K. Tanaka. Zero-knowledge protocols for NTRU: Application to identification and proof of plaintext knowledge. In *ProvSec 2009*.